Quelques aspects algorithmiques de la cryptographie

MÉMOIRE

présenté

devant l'Université de Rennes 1

pour l'obtention du

Diplôme d'Habilitation à Diriger des Recherches
Mention Mathématiques et Applications

par

David Lubicz

Institut de Recherche Mathématique de Rennes
École Doctorale Matisse
U.F.R. de Mathématiques

Soutenue le 14 Novembre 2008 devant la commission d'examen

Rapporteurs	M.	Gerhard	Frey	Université de Duisburg-Essen
	M.	Guillaume	Hanrot	INRIA
	M.	Andreas	Stein	Université d'Oldenbourg
Examinateurs	M.	Antoine	Chambert-Loir	Université de Rennes I
	M.	Jean-Marc	Couveignes	Université Toulouse II
	M.	Jean-François	Mestre	Université de Paris Jussieu
	Mme.	Marie-Françoise	Roy	Université de Rennes I

Remerciements

Je voudrais tout d'abord remercier Gehrard Frey, Guillaume Hanrot ainsi qu'Andreas Stein qui m'ont fait l'honneur de rapporter ce mémoire. Je fais aussi part de ma gratitude envers Antoine Chambert-Loir, Jean-Marc Couveignes, Jean-François Mestre ainsi que Marie-Françoise Roy qui ont accepté de faire partie de mon jury.

Ce mémoire n'aurait sans doute pas vu le jour sans Reynald Lercier, qui m'a beaucoup aidé à retrouver une activitée de recherche productive à un moment où, quelque peu désorienté par un changement thématique, je me contentais de satisfaire à ma curiosité. Qu'il en soit remercié.

Le métier de chercheur serait d'une grande austérité s'il ne tenait pour beaucoup de l'émulation, du regard croisé et du travail partagé. Je profite de cette occasion qui m'est donnée pour dire le plaisir que j'ai eu à travailler avec mes coauteurs Robert Carls, Gweltaz Chatel, Jean-Charles Faugère, Pierrick Gaudry, David Kohel, Reynald Lercier et Thomas Sirvent.

Une partie importante de mon mémoire est inspirée d'une idée merveilleuse de Jean-François Mestre, qui a eu la gentillesse d'en partager la primeur lors d'un de nos séminaires de cryptographie. Je lui fais part de ma reconnaissance.

Je me sens aussi redevable envers Marie-Françoise Roy qui a beaucoup œuvré pour le développement de l'activité scientifique autour de la cryptographie au sein de l'Institut de Recherche en Mathématiques de Rennes et par là-même me permettre de bénéficier d'excellentes conditions de travail. Je voudrais de même souligner le rôle important joué par Michel Coste qui nous a généreusement accueillis et pleinement intégrés dans son équipe, Reynald Lercier, Pierre Loidreau et moi-même.

Je ne saurais terminer ces remerciements sans rappeler à quel point j'ai plaisir à travailler au sein du laboratoire de cryptographie du CELAR. D'une part, j'aime mon métier d'ingénieur qui maintient une tension, à mon avis, enrichissante sur mon activité de recherche. D'autre part, je me sens très obligé de la stimulation et de la bonne humeur que me procurent les membres de l'équipe à savoir Didier Alquié, Stéphanie Alt, Johann Barbier, Didier Burel, Vivien Dubois, Franck Landelle, Reynald Lercier, Pierre Loidreau, Emmanuel Mayer, Thomas Sirvent ainsi que Valérie Rouat.

Table des matières

Chapitre 1

Introduction

L'objet de ce chapitre est de montrer comment l'ensemble des travaux que nous présentons s'organisent autour de questions classiques liées à la cryptographie et l'algorithmique. Son découpage préfigure de l'organisation générale de ce mémoire.

1.1 Protocoles cryptographiques et logarithme discret

Le point central qui constitue le liant entre les différents sujets abordés dans ce document est l'étude du problème du logarithme discret et de son utilisation dans des protocoles cryptographiques. Nous allons donc en rappeler brièvement le principe. Soit G un groupe cyclique d'ordre n et soit g un générateur de G. On utilisera la notation multiplicative pour la loi de groupe de G. Si x est un élément de G, le logarithme discret de x en base g est par définition l'unique entier noté $\log_g x$ tel que $0 \leq \log_g x < n$ et $g^{\log_g x} = x$. Ainsi la fonction logarithme discret est un inverse à droite de la fonction exponentielle $x \mapsto g^x$.

L'intérêt de la fonction logarithme discret en cryptographie ne peut se comprendre que si on l'interprète dans le langage de la théorie de la complexité. On se donne alors $(G_\alpha)_{\alpha \in \mathscr{P}}$ une famille de groupes cycliques dépendant d'un ensemble de paramètres $\mathscr{P}$. Dans tout ce qui suit le paramètre de complexité $n(\alpha)$ servant à évaluer le temps de calcul des algorithmes est le logarithme de l'ordre du groupe G_α. On suppose que l'ensemble $\{n(\alpha), \alpha \in \mathscr{P}\} \subset \mathbb{N}$ est non borné. Pour chaque $\alpha \in \mathscr{P}$, l'ensemble sous-jacent au groupe G_α est un ensemble fini de mots de longueur de l'ordre de celle de $n(\alpha)$ et la loi de groupe $+_\alpha$ est donnée par un algorithme s'exécutant en temps polynomial en $n(\alpha)$. Le groupe G_α vient avec des éléments distingués, à savoir l'élément neutre 0_α et un générateur g_α. Avec ces données, il est possible de se poser la question de la difficulté asymptotique du problème du logarithme discret, c'est-à-dire quelle est la complexité du meilleur algorithme résolvant le problème du logarithme discret dans la famille $(G_\alpha)_{\alpha \in \mathscr{P}}$.

De manière plus complète, pour faire fonctionner de nombreuses primitives cryptographiques, il est nécessaire de trouver au préalable une famille de groupes cycliques finis $(G_\alpha)_{\alpha \in \mathscr{P}}$ dont la taille est une fonction non bornée de α ayant les propriétés suivantes :

- la loi de groupe de G_α se calcule en temps polynomial ;
- le problème du logarithme discret dans G_α est difficile, i.e. non polynomial ;
- il est possible de calculer en temps polynomial la cardinalité de G_α ;
- si l'on considère un sous-ensemble de paramètres α correspondant à des groupes G_α de taille bornée, il est possible de tirer un élément de cet ensemble selon une distribution uniforme en temps polynomial.

En fait, il y a bien peu de familles de groupes dont on pense qu'elles vérifient l'ensemble de ces hypothèses. Les plus utilisées sont les groupes des éléments inversibles d'un corps fini, les groupes des points rationnels des courbes elliptiques définies sur un corps fini ou plus généralement les groupes des points rationnels des courbes algébriques définies sur un corps fini.

L'idée derrière ces hypothèses est que si elles sont vérifiées alors la fonction exponentielle $x \mapsto g^x$ est une fonction à sens unique ce qui constitue une brique élémentaire fondamentale en cryptographie. Par exemple, à partir d'une fonction à sens unique il est possible de construire un générateur de pseudo-aléa et par là une primitive de chiffrement sémantiquement sûre [86] dont la sécurité repose sur la difficulté à inverser la fonction à sens unique. Les hypothèses sus-citées constituent donc une préoccupation importante pour la cryptographie et permettent de motiver la plupart des travaux qui sont présentés dans ce mémoire.

Il convient de noter que l'existence d'une fonction à sens unique implique facilement que $P \neq NP$. Cette inégalité constitue une des plus importantes conjectures de la théorie de la complexité et elle reste pour l'instant un mystère. Une conséquence de cela est qu'à ce jour, nous ne disposons d'aucune certitude sur la sécurité des primitives reposant sur le problème du logarithme discret. Ce que l'on peut dire est que le problème du logarithme discret dans les corps finis et les courbes elliptiques définies sur les corps finis est très étudié et que jusqu'à présent personne n'a trouvé d'algorithme de complexité polynomiale pour le résoudre. Cette situation n'est pas vraiment satisfaisante et nous aimerions disposer aussi d'arguments de sécurité positifs. En fait, une idée possible pour essayer d'évaluer la complexité du problème du logarithme discret est d'affaiblir le modèle de calcul que l'on utilise à savoir celui donné par les machines de Turing. C'est suivant cette idée que Nechaev et Shoup [100, 115], ont proposé le modèle du groupe générique. De manière imagée, le modèle du groupe générique est donné par un automate à mémoire dont les seules opérations autorisées sont les opérations de groupe. L'article de Shoup montre que dans ce modèle le meilleur algorithme pour résoudre le problème du logarithme discret, par exemple "Pas de bébé-Pas de Géant", est de complexité exponentielle en la taille des éléments du groupe. Il en est de même pour les problèmes apparentés au logarithme discret comme le problème Diffie-Hellman calculatoire et décisionnel. Dans la version du modèle du groupe générique décrite dans les articles [100, 115], il est difficile de comparer le modèle de calcul du groupe générique avec le modèle de calcul de référence en théorie de la complexité qui est celui d'une machine de Turing. C'est pour cela que certains auteurs ont amélioré la présentation initiale. Le modèle du groupe générique n'est en fait rien d'autre qu'une machine de Turing qui doit résoudre un problème relatif à un groupe que l'algorithme ne peut manipuler que par l'intermédiaire d'oracles qui tirent le groupe et ses lois dans une famille de manière aléatoire.

Le modèle du groupe générique est maintenant considéré comme un outil standard en cryptographie. Il sert soit à prouver une primitive [14, 15, 83] ou plus généralement à évaluer la sécurité d'un problème sur lequel repose une primitive [7, 8]. Le modèle du groupe générique a été étendu en conséquence par certains auteurs afin de tenir compte des couplages [8]. En effet, dans [53], Joux a montré l'intérêt que représente l'opération de couplage de Weil ou de Tate calculable efficacement sur certaines courbes algébriques comme brique de base pour la construction de nouveaux protocoles. L'utilisation des couplages a donné lieu à une intense activité scientifique dans le domaine des protocoles cryptographiques dont les réalisations les plus notoires sont le chiffrement basé sur l'identité, des protocoles de diffusion, des signatures courtes, des signatures prouvées dans le modèle standard.

Du fait de sa place centrale comme modèle pour faire des preuves cryptographiques, le modèle du groupe générique a été l'objet de nombreuses critiques [119, 32], la pertinence de certaines d'entre elles ayant été discutés [57]. Une des critiques est que dans les groupes réels, la loi de groupe est facilement distinguable d'un oracle qui se contenterai de tirer un motif aléatoire à chaque requête.

Dans l'article [84], nous développons un modèle du groupe générique avec couplage qui est le plus réaliste possible. En particulier, l'attaquant a la possibilité de faire des calculs sur les motifs obtenus à partir des oracles de groupes et de re-soumettre le résultat pour une nouvelle opération de groupe. Notre point de vue est différent de celui adopté dans les présentations habituelles du groupe générique. En effet, dans les travaux de Shoup l'oracle de groupe répond systématiquement de manière aléatoire à une requête fraîche. Ce n'est qu'à la fin du jeu qu'une loi de groupe est tirée au hasard et que le jeu avec la vrai loi de groupe est comparé avec le jeu simulé par des tirages aléatoires. Dans notre modèle, l'oracle tire au début une loi de groupe et répond suivant cette loi pendant toute la durée du jeu. Nous calculons une probabilité de collision et montrons que cela donne une majoration pour le gain de l'adversaire.

Nous utilisons de manière essentielle ce point de vue afin d'introduire la notion de groupe pseudo-aléatoire qui est une généralisation de la notion de groupe générique. De manière intuitive, un groupe pseudo-aléatoire est un groupe dans lequel la loi de groupe est tirée suivant une distribution fortement pseudo-aléatoire (le mot "fortement" signifie que dans la définition du modèle de sécurité l'attaquant a accès à la fonction choisie dans une famille ainsi qu'à son inverse). Le résultat principal de [84] montre que tout résultat qui peut être prouvé dans le modèle du groupe générique peut aussi être prouvé dans le modèle du groupe pseudo-aléatoire. En d'autres termes, il est possible de considérer des familles de groupes pour lesquelles les lois de groupe simulées par des oracles sont tirées non pas de manière aléatoire mais suivant une distribution quelconque. La sécurité d'une hypothèse sur une telle famille peut se réduire sur la sécurité de cette famille considérée comme une famille de groupes pseudo-aléatoires à la sécurité de la même hypothèse dans le modèle du groupe générique. Ce résultat montre que la notion de groupe pseudo-aléatoire ne permet pas de dégager une notion de sécurité qui est différente de celle que l'on obtient avec le modèle du groupe générique. Il n'en reste pas moins vrai que le modèle des groupes pseudo-aléatoires montre qu'il est possible de gagner en réalisme sur les présentations classiques du modèle du groupe générique.

Nous avons par la suite appliqué le modèle du groupe générique à la preuve d'un protocole de diffusion dont le comportement en terme de performance dans des scénari d'utilisation réalistes améliore sensiblement celui des protocoles équivalents connus. Un protocole de diffusion [37] est utilisé lorsqu'un émetteur souhaite envoyer des messages à plusieurs destinataires par le biais d'un canal non sécurisé. Un tel schéma permet à l'émetteur de choisir de manière dynamique un sous-ensemble appelé "utilisateurs privilégiés" dans l'ensemble de tous les destinataires possibles et d'envoyer un chiffré lisible uniquement par les utilisateurs privilégiés. Ce genre de schéma est très utile dans de nombreuses applications commerciales comme la télévision à péage ou la diffusion de contenus multimédias. Bien sûr, ce service peut être obtenu avec des protocoles de chiffrement classiques, mais au prix d'une explosion de la taille des chiffrés.

De nombreux schémas ont été proposés pour résoudre ce problème au regard de deux principaux aspects. Dans une première approche, il s'agit de gérer un ensemble presque fixe d'utilisateurs. Dans ce cas, le chiffrement est très efficace mais pour modifier l'ensemble des utilisateurs privilégiés, il est nécessaire d'envoyer un long message. La deuxième approche est destinée à la gestion d'un très grand nombre d'utilisateurs privilégiés. Les schémas conçus dans ce but permettent de changer sans sur-coût l'ensemble des utilisateurs privilégiés mais la taille des chiffrés crois linéairement avec la taille de l'ensemble des utilisateurs révoqués.

Dans l'article [83], nous considérons une application réelle dans laquelle un émetteur produit différents types de contenus pour des catégories distinctes d'utilisateurs. Cela correspond à une problématique naturelle pour un opérateur qui utilise un schéma de diffusion asymétrique. Dans ce cas, il est tout à fait possible que l'ensemble des utilisateurs privilégiés change de manière importante en même temps que le type du contenu. Comme cet ensemble ne peut pas être considéré comme étant particulièrement petit ou grand, cette situation n'est pas correctement couverte par les schémas de diffusion habituels.

Récemment, une notion de chiffrement basé sur les attributs a été introduite [109]. Cette notion permet de résoudre le genre de problème qui nous intéresse. Dans [51], les auteurs présentent une déclinaisons de ces idées avec une application dans ce qui est appelé un schéma de diffusion ciblé. Dans les schémas de type "ciphertext-policy", qui nous préoccupent ici, un utilisateur est associé à un ensemble d'attributs et sa clef de chiffrement dépend de cet ensemble. Un chiffré contient une clef d'accès calculée à partir de ces attributs : seuls les utilisateurs qui satisfont à cette politique peuvent déchiffrer et toute collusion d'autres utilisateurs ne peut obtenir le clair. Ici, par politique d'accès, on entend une fonction à valeur booléenne calculée à partir des attributs. Dans des applications de diffusion, le principal inconvénient de cette famille de schémas est que le déchiffrement requiert d'importants calculs qui sont difficiles à effectuer sur des décodeurs à bas coût.

Dans le papier [83], nous proposons un schéma de diffusion avec un mécanisme de type attribut : il permet à un émetteur de sélectionner ou de révoquer non seulement des utilisateurs isolés mais aussi des groupes d'utilisateurs définis par leurs attributs.

Ce mécanisme peut être vu comme un schéma basé sur les attributs avec un procédé de déchiffrement et une politique d'accès particulièrement efficace : la politique d'accès (utilisant des fonction AND ou NOT) est suffisante pour un schéma de diffusion puisque la fonctionnalité OU peut être simulée par des concaténations de chiffrés exactement comme dans le paradigme des recouvrements de sous-ensembles.

L'idée derrière ce schéma est la possibilité de calculer le plus grand diviseur commun de deux polynômes cachés par une exponentiation dans un groupe. Chaque récepteur est associé à un polynôme (dont les racines dépendent de ses attributs) et un chiffré est associé à un autre polynôme (dont les racines dépendent des attributs requis et des attributs révoqués). Un récepteur appartenant à la politique d'accès définie par un chiffré calcule le plus grand diviseur commun de son propre polynôme et du polynôme associé au chiffré : ce diviseur est le même pour tous les récepteurs dans la politique d'accès. Un récepteur n'appartenant pas à la politique d'accès obtiendrait un polynôme différent : ou bien ce polynôme ne peut être calculé, ou bien il ne peut pas être utilisé pour calculer le chiffré.

Dans ce schéma, la taille de la clef de déchiffrement donnée au récepteur est linéaire en le nombre des attributs qui lui sont associés. La taille d'un chiffré est linéaire en le nombre d'attributs utilisés dans la politique d'accès. La clef de chiffrement publique est longue : sa taille est linéaire en le nombre total d'attributs utilisés dans le schéma. Cela ne présente pas un réel inconvénient dans des situations réalistes pour lesquelles de toute manière l'émetteur doit disposer d'une base de donnée contenant la liste des usagés avec leurs attributs. De plus, un émetteur qui a l'intention d'utiliser seulement une petite fraction de l'ensemble des attributs n'a besoin que des clefs de chiffrement dont la taille est linéaire en la taille de ce petit ensemble.

Le schéma décrit dans ce papier ne repose pas sur du partage de secret comme dans le cas des schémas basés sur les attributs déjà connus. Il est conçu de manière à ce que le mécanisme de déchiffrement n'utilise qu'un nombre constant de calculs de couplages.

Comme schéma de diffusion, il utilise le paradigme des recouvrements de sous-ensembles suggéré dans [99]. Nous prouvons que le schéma est sûr contre une collusion pleine dans le modèle du groupe générique avec couplage.

Une propriété intéressante de notre schéma est qu'une clef de déchiffrement nouvelle peut être construite sans modification des clefs de déchiffrement précédemment distribuées : ajouter une nouvelle clef de déchiffrement requiert seulement d'étendre la clef publique pour tenir compte de nouveaux attributs.

1.2 Des algorithmes de comptage de points

Nous avons vu dans les paragraphes précédents que l'infrastructure générale donnée par le problème du logarithme discret permet de construire de nombreuses variantes de protocoles ayant des propriétés ajustables en fonction des besoins. Dans ce paragraphe, nous revenons sur les groupes disponibles pour instancier le problème du logarithme discret. Au début de cette synthèse, nous avons listé un certain nombre de conditions que doit vérifier une famille de groupes pour constituer un candidat acceptable. Une de ces conditions est qu'il n'existe pas d'algorithme efficace pour résoudre le problème du logarithme discret dans cette famille. Cela élimine en particulier tous les groupes abéliens donnés sous la forme présentation-relation. En fin de compte, il n'existe pas beaucoup de familles utilisables en cryptographie. Pour l'essentiel, les deux familles les plus maniables sont les groupes des éléments multiplicatifs d'un corps fini et les groupes donnés par les points rationnels des jacobiennes de courbes algébriques projectives lisses définies sur un corps fini. Dans le premier cas, nous rappelons que les meilleurs algorithmes connus pour attaquer le logarithme discret sont sous-exponentiels. Le reste de ce paragraphe est consacré à l'étude du deuxième cas.

La première famille de courbes à laquelle on peut penser pour engendrer des groupes sont les courbes elliptiques. En effet, l'ensemble de leurs points rationnels forme déjà un groupe dans lequel il est aisé de calculer. Afin d'éliminer les groupes qui correspondent à des instances triviales du problème du logarithme discret, il est nécessaire de pouvoir calculer leur ordre. En effet, dans le cas d'une famille de courbes elliptiques ayant un ordre n friable c'est-à-dire que le plus grand diviseur premier de n est de l'ordre de $\log(n)$, le classique algorithme de Pohlig-Hellman [92] permet d'attaquer le problème du logarithme discret en temps polynomial. Cette préoccupation a été à l'origine de très nombreux travaux afin d'améliorer les algorithmes de comptage de points rationnels dans les courbes algébriques définies sur un corps fini. Nous en rappelons brièvement l'historique.

Dans ce qui suit, nous fixons un corps de base fini $\mathbb{F}_q$ à $q = p^n$ éléments ; p premier. Le premier algorithme de calcul du nombre de points rationnels d'une courbe elliptique définie sur un corps fini en temps polynomial est dû à Schoof [112]. Cet algorithme a été par la suite amélioré par Elkies et Atkin [33] pour obtenir un temps d'exécution en $O(\log q)^{2\mu+2}$. Dans cette dernière formule, μ est une constante telle que le produit de deux entiers de longueur n peut être fait en $O(n^\mu)$ opérations élémentaires. Par exemple, si on utilise l'algorithme trivial $\mu = 2$ tandis qu'avec la transformée de Fourrier rapide, on a $\mu = 1 + \epsilon$, ϵ étant un terme logarithmique. Nous utiliserons librement dans ce mémoire la notation dite O-souple qui consiste à négliger les termes logarithmiques dans les estimations de complexité. Ainsi, avec cette notation, la complexité de l'algorithme de Schoof, Elkies et Atkin (SEA) s'écrit $\tilde{O}(n^4)$. L'algorithme SEA est le seul algorithme connu pour calculer le nombre de points d'une courbe elliptique ayant un comportement polynomial en fonction de la caractéristique du corps de base.

L'algorithme de Schoof a été généralisé dans le cas des courbes hyperelliptiques par Pila dans [106]. Bien que de complexité polynomiale, l'algorithme de Pila ne bénéficie pas des améliorations dues à Elkies et Atkin. Cependant, il est possible de rendre l'algorithme de Pila plus rapide [44] ce qui permet de calculer le nombre de points rationnels d'une courbe hyperelliptiques sur un corps fini de taille presque cryptographique [48].

En dehors de ces algorithmes ℓ-adiques, une autre importante famille d'algorithmes utilise des techniques p-adiques. L'idée commune de tous ces algorithmes est de construire un relevé du morphisme de Frobenius sur un corps de caractéristique zéro. On peut les diviser en deux grandes familles :

- Les méthodes cohomologiques qui consistent à calculer l'action du morphisme de Frobenius sur des groupes de cohomologie définis sur un corps de caractéristique

0. Ces méthodes reposent soit sur la cohomologie de Monsky-Washnitzer suivant Kedlaya [54] et ses améliorations [31, 30, 29, 49], soit sur la cohomologie de Dwork [65, 66, 63, 69, 64, 62, 68]. Ces techniques donnent des algorithmes de complexité cubique pour des classes de courbes très générales.
– Les méthodes de type relèvement qui calculent l'action du morphisme de Frobenius sur le relevé canonique de la Jacobienne d'une courbe hyperelliptique. Cela donne des algorithmes très efficaces pour les courbes elliptiques et les courbes hyperelliptiques.

Le premier algorithme p-adique fut proposé par Satoh [111]. L'idée de cet algorithme est de calculer le relevé sur les p-adiques d'une courbe elliptique E_k définie sur un corps fini de petite caractéristique p de manière à ce que l'anneau d'endomorphismes du relevé soit le même que celui de E_k. Un tel relevé existe toujours si E_k est ordinaire et s'appelle le relevé canonique de E_k. Si l'on fixe p, la complexité en temps de l'algorithme de Satoh, lorsque l'on fait tendre le degré n de l'extension k sur $\mathbb{F}_p$ est $O(n^{2\mu+1})$ et sa complexité en mémoire est $O(n^3)$. Peu de temps après, Vercauteren, Preneel, Vandewalle [124] on réduit la complexité en mémoire à $O(n^2)$. De manière indépendante, dans le cas de la caractéristique 2, un algorithme utilisant le calcul itéré de la moyenne arithmético-géométrique (AGM) ayant même complexité asymptotique que l'algorithme de Satoh fut découvert par Mestre [93].

Dans la suite, nous décrivons nos contributions dans le domaine des algorithmes de comptage de points. Elles consistent principalement en

– une amélioration algorithmique de l'algorithme de Mestre qui permet d'obtenir un algorithme de complexité quasi-quadratique en temps et quadratique en espace [77, 79] ;
– une généralisation de l'algorithme de Mestre et Satoh qui sont maintenant englobés dans un même algorithme qui permet de calculer le nombre de points d'une courbe hyperelliptique générale en temps quasi-quadratique et ce quelque soit la caractéristique du corps de base [17] ;
– un algorithme de comptage de points qui utilise la cohomologie de Monsky-Washnitzer à support propre [22].

L'algorithme de Satoh ainsi que celui de Mestre sont très proches, le second pouvant être considéré comme une interprétation élégante du premier. Ils partagent en particulier la même structure algorithmique qui se décompose en deux phases :

– le calcul d'un relèvement sur les p-adiques d'invariants modulaires associés à la jacobienne de la courbe que l'on considère ;
– le calcul de l'action du morphisme de Frobenius ou plutôt de son dual, sur l'espace des formes différentielles invariantes du relevé canonique.

La deuxième étape de l'algorithme consiste principalement en un calcul de norme que l'on peut faire en temps quasi-quadratique en la taille du corps de base. L'objet de l'article [77] est de montrer qu'il est aussi possible d'effectuer la première étape en temps quasi-quadratique.

Dans le cas de l'algorithme de Satoh, d'après un théorème de Lubin-Serre-Tate, cette première étape revient à relever sur les p-adiques une certaine équation modulaire. Plus précisément, si E_k est une courbe elliptique ordinaire sur un corps fini k de j-invariant j

alors le j-invariant de son relevé canonique est uniquement défini par les deux conditions

$$\Phi_p(X, \Sigma(X)) = 0 \qquad \text{et} \qquad X = j \mod p,$$

où Σ est le relevé du morphisme de Frobenius sur les p-adiques et $\Phi_p(X, Y)$ est le polynôme modulaire d'ordre p [117, p. 181].

Dans le cas de l'algorithme de Mestre, il s'agit de résoudre une équation de la forme

$$\Sigma(X) = \frac{2\sqrt{X}}{1+X}.$$

On voit donc que le problème qui nous intéresse ici est le calcul d'un relevé sur les p-adiques d'une solution connue à petite précision d'une équation polynomiale tordue par l'action du morphisme de Frobenius. Il existe un algorithme bien connu et très efficace sur les p-adiques pour relever des solutions d'équations polynomiales dont on connaît une approximation à petite précision : c'est l'algorithme de Newton qui permet de doubler la précision p-adique connue à chaque itération. Le problème est qu'il n'est pas possible d'appliquer l'algorithme de Newton dans notre cas puisque nous n'avons pas à faire à des équations polynomiales.

L'idée principale de [77] est de montrer qu'il est possible de généraliser l'algorithme de Newton aux polynômes tordus par le Frobenius si l'on arrive à résoudre efficacement une équation de la forme :

$$\Sigma(X) + aX + b = 0,$$

a et b étant des nombres p-adiques, que nous avons appelé équation d'Artin-Schreier généralisée.

Nous donnons un algorithme pour résoudre efficacement de telles équations. L'algorithme tire partie d'une loi de composition sur des relations de la forme $\Sigma^k(X) + a_k X + b_k = 0$ pour effectuer une boucle inspirée de l'algorithme d'exponentiation rapide. Dans le cas où le corps de base admet une base Gaussienne optimale, on obtient alors un algorithme de relèvement de complexité quasi-quadratique. La phase de norme étant aussi quasi-quadratique dans ce cas, l'algorithme de comptage de points ainsi obtenu est de complexité quasi-quadratique.

Il convient de remarquer que ce résultat constitue une sorte d'optimum pour le problème du comptage de points sur une courbe algébrique : en effet, la vérification du calcul qui consiste pour l'essentiel à prendre un point de la courbe au hasard et à l'élever à la puissance l'ordre supposé du groupe ne peut se faire au mieux qu'en temps quadratique. Ainsi, dans les exemple de gros calculs que nous avons effectués, l'étape de vérification, qui n'était pas optimisée, est celle qui a pris le plus de temps. Cela illustre l'importance des progrès qui ont été réalisés dans le domaine du comptage points : en effet, il y a quelques années, avant la découverte de l'algorithme de Schoof, compter le nombre de points rationnels sur une courbe algébrique définie sur un corps fini était considéré comme un problème difficile.

Une implémentation soignée en langage C de cet algorithme nous a permis d'effectuer des calculs sur des corps de base bien plus grands que ceux en usage en cryptographie [75, 73, 74].

Dans [94], Mestre remarque qu'il est possible d'interpréter l'AGM comme une forme particulière des formules de duplication de Riemann pour les fonctions thêta. Ces formules permettent en particulier de relier les valeurs de thêta constantes associées à des variétés abéliennes 2^g-isogènes, g étant la dimension des variétés abéliennes. Cela permet de donner une généralisation de l'AGM pour des courbes hyperelliptiques.

Cet algorithme délicat utilise de nombreux résultats mathématiques et algorithmiques. Dans [79], nous décrivons une amélioration de cet algorithme avec une complexité en temps quasi-quadratique. L'apport de cet article est double :

- nous présentons une version vectorielle de l'algorithme de résolution rapide d'équations polynomiales tordues par le morphisme de Frobenius ;
- nous donnons une preuve de correction de l'algorithme en exhibant un équivalent en genre plus grand que 1 de l'équation modulaire utilisée dans l'algorithme de Satoh pour relever canoniquement le j-invariant.

Cet algorithme constitue une illustration particulièrement frappante d'application des techniques dites transcendantes en arithmétique. L'idée est d'utiliser le plongement des p-adiques dans le corps des nombres complexes d'une part, d'autre part la théorie analytique et en particulier le théorème d'uniformisation des variétés analytiques complexes pour construire des relations algébriques ayant un sens arithmétique. D'ailleurs, les seules preuves connues de certaines formules essentielles à l'algorithme de Mestre utilisent la géométrie analytique complexe. C'est le cas de la formule de Thomae-Fay qui peuvent se démontrer par des techniques de déformation de variétés analytiques complexes. Ces considérations donnent une idée des liens que peuvent avoir des travaux géométriques [81, 80] avec les résultats algorithmiques que nous présentons dans ce mémoire.

Là encore, une implémentation soigneuse de cet algorithme nous a permis d'effectuer des calculs sur des courbes de genre 2 pour des tailles de corps de base bien plus grandes que celles nécessaires dans des applications cryptographiques [72, 76, 78]. Le calcul en genre 2 sur un corps de taille 32770 bits constitue le plus grand calcul de ce type effectué à ce jour.

L'amélioration décrite dans l'article [79] montre l'existence d'un algorithme de complexité quasi-quadratique afin de calculer le nombre de points d'une courbe hyperelliptique définie sur un corps de caractéristique deux. De manière plus générale, on peut se poser la question de savoir s'il existe un algorithme de complexité quasi-quadratique à genre et caractéristique fixée pour compter le nombre de points d'une courbe algébrique quelconque.

L'algorithme de Mestre montre que la réponse à cette question est probablement affirmative. Encore, faut-il trouver une manière des généraliser les équations modulaires utilisées dans cet algorithme. C'est ce que nous faisons dans l'article [17]. Le résultat principal de cet article montre qu'il existe un algorithme de complexité quasi-quadratique à genre et caractéristique fixée afin de compter le nombre de points d'une courbe hyperelliptique. En fait, dès que l'on dispose de certains invariants modulaires associés à une variété abélienne définie sur un corps fini, il est possible de calculer sa fonction zêta associée de manière très efficace.

Notre algorithme reprend l'idée générale de Satoh/Mestre. Il s'agit de relever canoniquement certains invariants modulaires associés à la jacobienne de la courbe de départ, puis de calculer l'action du morphisme de Frobenius sur l'espace des formes différentielles invariantes du relevé canonique. Les invariants que nous utilisons sont les thêta constantes. Pour cela, nous utilisons de manière essentielle une théorie arithmétique des fonctions thêtas développée par Mumford dans [95] après des idées de Weil et Igusa. L'idée principale de cette théorie est que les fonctions thêta ne sont rien d'autre que la donnée d'une base particulière des sections globales d'un fibré ample sur une variété abélienne. Cette base est en fait définie dès que l'on connaît l'action d'un certain groupe appelé groupe thêta sur les sections globales du fibré ample. L'intérêt de cette théorie est qu'elle permet d'utiliser la plupart des résultats de la théorie géométrique dans un contexte arithmétique ce qui est important lorsque l'on veut faire intervenir le morphisme de Frobenius dans des équations modulaires.

L'algorithme décrit dans l'article [17] utilise de manière essentielle les résultats originaux suivants :

- des équations modulaires qui définissent le relevé canonique d'une variété abélienne dans l'espace de déformation local décrit dans le système de coordonnées des thêta constantes ;
- une formule de transformation généralisée qui permet de calculer l'action du morphisme de Frobenius sur les formes invariantes du relevé canonique de la jacobienne de la courbe considérée à partir de la connaissance des thêta constantes associées à une structure de niveau la caractéristique du corps résiduel.

Cette dernière formule utilise le fait que les thêta constantes sont des formes modulaires sur le demi-espace de Siegel.

Nous avons fait une implémentation de cet algorithme en magma et écrit une librairie qui permettent de calculer les équation modulaires que nous utilisons.

Nous avons vu que les algorithmes de comptage de points peuvent être considérés comme le pendant algorithme d'une théorie cohomologique ayant de bonnes propriétés. Ces cohomologies dites de Weil ont été un sujet d'intérêt majeur des mathématiques de l'après guerre dans le sillage des travaux de Serre et Grothendieck en vue de la démonstration des conjectures de Weil. De nombreux candidats pour des cohomologies de Weil ont été décrits à cette époque et ils ont été revisités ces dernières années afin de concevoir des algorithmes de comptage de points [54, 63, 66, 69].

L'article [22] s'inscrit dans cette veine. Nous y présentons un algorithme de comptage de points utilisant la cohomologie de Monsky-Washnitzer à support propre. Cette cohomologie est munie d'une formule des traces de Lefschetz et donne donc lieu à un algorithme de comptage efficace. L'idée de la théorie de Monsky et Washnitzer est que pour obtenir une cohomologie dans un corps de caractéristique 0, il est possible de relever la courbe de départ disons V_k définie sur un corps fini k dans les p-adiques. Notons V ce relevé. Le problème est qu'en général, il n'est pas possible de relever le morphisme de Frobenius de V_k dans V. Pour cela, il est nécessaire de considérer le schéma formel $\tilde{V}$ associé à V qui peut être vu comme un épaississement infinitésimal de V. Il est alors possible de relever le morphisme de Frobenius à $\tilde{V}$ et de manière usuelle, on peut définir une cohomologie de de Rham dans $\tilde{V}$. Mais là encore une difficulté survient : les groupes de cohomologie ainsi obtenus ne sont pas en général de dimension finie. L'idée de Monsky et Washnitzer est de ne regarder qu'un sous-espace de l'espace des fonctions définies sur $\tilde{V}$ à savoir celui des fonctions dites sur-convergentes.

Dans [54], Kedlaya a montré qu'il était possible d'utiliser la cohomologie de Monsky-Washnitzer afin de faire du comptage de points. Son algorithme est relativement simple. Dans le cas considéré des courbes hyperelliptiques, il exhibe une base de la cohomologie et explique comment il est possible de relever le morphisme de Frobenius et de calculer son action sur la base décrite. Le plus dur est alors d'évaluer la précision analytique nécessaire pour être sûr que le résultat final est une approximation p-adique suffisante du polynôme caractéristique du morphisme de Frobenius.

Notre algorithme a la même structure que celui de Kedlaya et peut donc être décomposé de deux parties. La première consiste en le calcul d'une base de la cohomologie de Monsky-Washnitzer à support compact. Contrairement à ce qui se passe avec l'algorithme de Kedlaya, cette étape est non triviale d'un point de vue algorithmique. Les deux principales originalités de notre algorithme sont :

- L'utilisation du théorème de stabilité de la cohomologie de Monsky-Washnitzer à support propre par descente finie étale afin de réduire le calcul d'une base de la cohomologie au calcul des sections globales horizontales d'un isocristal sur la droite affine. Ces sections globales vérifient une équation différentielle qui n'est autre que la connexion de Gauss-Manin. Nous traduisons cette équation différentielle sous la forme d'un système linéaire. C'est ce que nous appelons la méthode globale.
- Malheureusement, la méthode globale est inefficace puisque qu'elle nécessite l'inversion d'une matrice dont la taille est de l'ordre de la précision analytique des calculs. Nous expliquons comment accélérer le calcul de la base en utilisant des équations différentielles inhomogènes locales déduites de la connexion de Gauss-Manin. Une fois que l'on a trouvé des solutions globales pour la connexion de Gauss-Manin à petite précision analytique, il est possible de les prolonger localement de manière efficace en utilisant ces équations différentielles. C'est ce que nous appelons la méthode locale.

La deuxième partie de l'algorithme est le calcul d'un relevé du morphisme de Frobenius ainsi que son action sur la cohomologie. Le calcul du relevé du morphisme de Frobenius que nous utilisons est assez standard et consiste principalement en une adaptation de [54]. Les choses changent cependant, pour ce qui est du calcul de l'action du morphisme de Frobenius et nous expliquons comment il est possible de déduire un algorithme efficace de la connaissance d'équations différentielles locales déduites de l'équation de Gauss-Manin tordue par le morphisme de Frobenius.

Nous donnons une preuve de correction de notre algorithme. Comme d'habitude pour des algorithmes reposant sur la cohomologie p-adique, la partie la plus délicate consiste en l'évaluation des précisions analytique et p-adique nécessaires pour pouvoir retrouver le polynôme caractéristique du morphisme de Frobenius. À cette fin nous utilisons une variante d'un résultat de Lauder [67]. Nous présentons une analyse de complexité détaillée de notre algorithme. Dans cette estimation, nous supposons que la caractéristique du corps de base est fixée et considérons des bornes de complexité quand le genre g et le degré absolu n du corps de base tendent vers l'infini. En utilisant la notation O-souple afin de négliger les termes logarithmiques, nous obtenons une complexité en temps de $\tilde{O}(g^4n^3)$ avec une consommation mémoire de $O(g^3n^3)$. Ces résultats de complexité reposent de manière essentielle sur une arithmétique rapide pour des anneaux de séries formelles développée dans une série d'articles [2, 13, 105, 12]. Nous remarquons que notre algorithme a la même complexité par rapport à n que l'algorithme de Kedlaya.

1.3 Quelques calculs avec des surfaces abéliennes

Les courbes elliptiques admettent une généralisation naturelle en dimension plus grande que 1 à savoir les variétés abéliennes. Une variété abélienne est par définition une variétés projectives lisse munies d'une structure de groupe. Il est assez naturel de se poser la question de savoir si les variétés abéliennes sur un corps fini constitue une famille de groupes utilisable pour le problème du logarithme discret. On peut par exemple commencer par regarder les jacobiennes de courbes hyperelliptiques qui sont des cas particuliers de variétés abéliennes. En effet, on sait que grâce à un algorithme dû à Cantor [16], il est possible de représenter le groupe des points d'une jacobienne de courbe hyperelliptique et de calculer facilement sa loi de composition.

Il convient alors de vérifier si le problème du logarithme discret est difficile dans la famille de jacobiennes de courbes hyperelliptiques. Une série de travaux [1, 47, 121, 45] montre qu'il existe des algorithmes plus rapides que les algorithmes génériques pour résoudre le problème du logarithme discret dans une jacobienne de courbe hyperelliptiques de genre plus grand que 4. Les résultats précédent ne s'appliquent pas au cas des variétés abéliennes de dimension 2 ou surfaces abéliennes. Il est possible de montrer que les surfaces abéliennes définies sur un corps fini sont généralement des jacobiennes de courbes hyperelliptiques et constituent donc une famille particulièrement intéressante de groupes utilisables en cryptographie. Ces considérations motivent les travaux décrits dans cette section.

La théorie de la multiplication complexe permet d'obtenir des méthodes efficaces pour calculer des variétés abéliennes définies sur un corps fini ayant un anneau d'endomorphismes prescrit. Dans le cas des courbes elliptiques, on part de $\mathcal{O}$ un ordre dans un corps quadratique imaginaire de discriminent D. Soit $h = h(D)$ le nombre de classes de $\mathcal{O}$. Il est bien connu [117, Ch.II] qu'il existe exactement h classes d'isomorphisme de courbes elliptiques avec multiplication complexe par $\mathcal{O}$. Soit j_i avec $i = 1, \dots h$ leur j-invariant. La méthode CM usuelle pour les courbes elliptiques consiste à calculer les j_i en utilisant une arithmétique à virgule flottante. Il est possible alors de retrouver le polynôme des classes de Hilbert

$$H_D(X) = \prod_{i=1}^{h}(X - j_i)$$

à partir de son approximation réelle, en utilisant le fait qu'il est à coefficients entiers. Il est habituel d'évaluer la complexité de cet algorithme par rapport à la taille de la sortie. Comme h croit de manière linéaire en fonction de D, un paramètre de complexité naturel pour cet algorithme est D. Il convient de noter que les courbes elliptiques construites à partir de la méthode CM ont un anneau d'endomorphismes avec discriminant petit et se distinguent de ce fait des courbes elliptiques générales. Même si pour l'instant, la littérature publique de décrit aucun algorithme pour attaquer des courbes ayant un anneau d'endomorphismes avec un petit discriminant, cela limite leur utilisation lorsqu'il s'agit de protéger des données extrêmement sensibles.

En 2002, Couveignes et Henocq [27] ont introduit une méthode de construction CM utilisant le relèvement p-adique de courbes elliptiques. L'idée est calculer un relevé, i.e. un relevé de la courbe sur un corps fini en caractéristique zéro, de manière à ce que la jacobienne de la courbe relevée soit a multiplication complexe. En fait, on relève des invariants géométriques de la courbe en utilisant des équations modulaires.

Dans l'article [19], nous présentons des formules qui peuvent être vues comme un analogue 3-adique des formules de Mestre. Contrairement à ces dernières, nos équations ne contiennent pas d'information sur l'action du relevé du Frobenius relatif sur la cohomologie. Afin de construire un relevé canonique, nous appliquons une version modifiée de l'algorithme de relèvement décrit dans [79].

Il existe une approche 2-adique pour le genre 2 de Gaudry et al. [42], qui utilise la classique correspondance de Richelot pour relever canoniquement. Cette dernière méthode s'applique seulement à un corps à multiplication complexe K dans lesquels le premier 2 est complètement scindé dans l'extension quadratique K/K_0, où K_0 est le sous corps réel de K. Pour tout autre corps à multiplication complexe K, la réduction de la courbe à multiplication complexe en 2 n'est pas ordinaire. Donc, il n'existe pas de courbe à multiplication complexe par K non ordinaire pour servir d'entrée à l'algorithme. La méthode présentée dans l'article [19] échange cette condition en 2 avec une condition analogue en 3. Donc la méthode CM résultante s'applique à une large classe de corps CM qui ne sont pas abordables avec la méthode 2-adique de [42].

Nous prouvons nos équations en utilisant la théorie des fonctions thêta algébriques développée par Mumford [95]. Dans la situation 3-adique arithmétique, nous utilisons un système de coordonnées sur le relevé canonique dont l'existence est prouvée dans [21]. Notre algorithme est prouvé en utilisant la théorie de Serre-Tate.

Après avoir passé en revue des techniques de construction de groupes pour faire du logarithme discret, nous nous intéressons maintenant à des algorithmes efficaces pour calculer dans de tels groupes. En effet, le calcul efficace de la loi de groupe pour les courbes elliptiques et les jacobiennes de courbes hyperelliptiques a des applications multiples en théorie algorithmique de nombres et en cryptographie. Le cas des courbes elliptiques a été largement étudié depuis des années. Cependant, des développements récents [5] montrent que le sujet n'est pas clos. De même, pour les courbes de genre 2, la littérature est plus récente et a moins d'ampleur. Cependant, il existe déjà une grande variété de systèmes de coordonnées, avec des variantes dépendant de la caractéristique du corps de base, de la classe particulière des courbes considérées, ou du coût relatif de la multiplication et des inversions dans le corps de base (voir par exemple [24, Chapitre 14]).

Il serait aussi intéressant de représenter et calculer avec des objets plus généraux comme par exemple les variétés abéliennes et les variétés de Kummer. Nous rappelons que une variété de Kummer K est la variété projective singulière obtenue comme le quotient d'une variété abélienne A par le morphisme d'inversion agissant sur elle. De manière évidente, la loi de groupe sur A ne passe pas au quotient. Cependant, si $\overline{P}$ est un point géométrique de K, pour tout entier positif, il est possible de définir un point $n\overline{P}$ sur K de la manière suivante : tout d'abord, relever $\overline{P}$ sur un point P de A, calculer nP et le projeter sur K. Le résultat de ce calcul ne dépend pas du relevé choisi de $\overline{P}$ dans A. L'opération obtenue s'appelle la loi de pseudo-groupe de K. De manière schématique, la variété de Kummer K contient une grande partie de l'information liée à la structure de $\mathbb{Z}$-module de A et cette structure de $\mathbb{Z}$-module est principalement ce qui est utilisé en cryptographie. En particulier, dans le cas où A est une courbe elliptique donnée par son équation de Weiestrass, sa variété de Kummer associée est connue comme sa forme de Montgomery. Dans cette représentation, un point est donné par son abscisse et des algorithmes sont connus pour calculer efficacement la loi de groupe.

Suivant Chudnovsky et Chudnovsky [23], Gaudry [41] a montré comment utiliser la théorie des fonctions thêta afin d'obtenir une loi de pseudo-groupe efficace dans le cas ou le corps de base est de caractéristique p impaire en faisant certaines hypothèses de rationalité. Les formules de pseudo-groupe sont déduites des classiques formules de duplication de Riemann pour les fonctions thêta par application du principe de Lefschetz puisque tout a bonne réduction modulo p.

Malheureusement, les formules de [41] ont mauvaise réduction modulo 2. L'objectif de l'article [43] est de donner des formules analogues valables quand le corps de base est de caractéristique 2. Une première question est de trouver une équation pour la quartique de Kummer en caractéristique 2. Sur un corps de caractéristique impaire, il est possible suivant [95] de retrouver un tel modèle simplement en cherchant des polynômes homogènes de degré 4 invariants pour l'action du groupe thêta. Ces techniques avec d'autres arguments géométriques ont été étendues pour couvrir le cas de la caractéristique 2 dans [60]. Nous retrouvons les mêmes équations que [60] mais en utilisant une approche différente. Notre point de vue afin de prouver tous nos résultats est de considérer une surface de Kummer définie sur le corps k comme la fibre spéciale d'un schéma de Kummer $\tilde{K}$ sur $W(k)$ où $W(k)$ désigne l'anneau de valuation discrète de caractéristique 0 ayant pour corps résiduel k. Nous obtenons les résultats en cherchant des formules en caractéristique 0 ayant bonne réduction sur la fibre spéciale. Cela nous permet d'interpréter les coefficients apparaissant dans les équations de K dans les termes du reste modulo 4 des thêta constantes de $\tilde{K}$. Cette identification est importante afin d'énoncer et de prouver les formules de pseudo-addition. De plus, il convient de noter que de cette manière, nous décrivons des invariants modulaires définis sur le corps k qui sont équivalents aux thêta constantes usuelles. Notre preuve repose sur la caractérisation du relevé canonique dans le système de coordonnées donné par une structure thêta canonique de niveau 2.

Une autre contribution de l'article [43], est une dérivation de formules qui sont similaires aux formules de genre 1 de [41] en caractéristique impaire ainsi que leurs équivalents en caractéristique 2. Dans le cas de la caractéristique impaire, on trouve des formules qui sont très similaires à celles associées à la forme de Montgomery mais avec des propriétés légèrement différentes. En caractéristique 2, nous retrouvons des formules déjà publiées par Stam [118].

Une autre application des fonctions thêta est le calcul d'un équivalent pour les variétés abéliennes de dimension plus grande que un des classiques polynômes modulaires $\Phi_\ell(X, Y)$. Nous rappelons que si j est le j-invariant associé à une courbe elliptique E_k définie sur un corps k alors les racines de $\Phi_\ell(j, X)$ donnent les j-invariants des courbes elliptiques qui sont ℓ-isogènes à E_k. Ces polynômes modulaires ont d'importantes applications algorithmiques. Par exemple, Atkin et Elkies (voir [33]) utilisent la paramétrisation modulaire des sous-groupes de ℓ-torsion des courbes elliptiques afin d'améliorer l'algorithme de comptage de points de Schoof. De la même manière, il est possible d'améliorer l'algorithme original de Satoh [124, 79, 40, 110] en résolvant sur les p-adiques une équation donnée par les polynômes modulaires $\Phi_p(X, Y)$.

Ce dernier algorithme a fait l'objet d'une généralisation dans [58] utilisant la notion de correspondance modulaire orientée. La courbe modulaire $X_0(N)$ paramétrise les classes d'isomorphisme de courbes elliptiques avec un sous-groupe de points de N-torsion. Par exemple, la courbe $X_0(1)$ est la droite des j-invariants. Soit p un entier premier à N. Une application rationnelle $X_0(pN) \to X_0(N) \times X_0(N)$ est appelée correspondance modulaire orientée si l'image de chaque point de $X_0(pN)$ représenté par une paire (E, G) où G est un sous-groupe d'ordre pN de E est un couple $((E_1, G_1), (E_2, G_2))$ avec $E_1 = E$ et G_1 l'unique sous-groupe d'indice p de G, $E_2 = E/H$ et H est l'unique sous-groupe d'ordre p de G. Dans le cas où la courbe $X_0(N)$ est de genre 0, la correspondance peut s'exprimer comme une équation de la forme $\Phi(X, Y) = 0$ décrivant une courbe isomorphe à $X_0(pN)$ dans le produit $X_0(N) \times X_0(N)$ Par exemple, si l'on considère la correspondance orientée $X_0(\ell) \to X_0(1) \times X_0(1)$ pour ℓ un nombre premier alors le polynôme définissant son image dans le produit est le polynôme modulaire $\Phi_\ell(X, Y)$.

Dans l'article [35], nous nous intéressons au calcul d'analogues des correspondances modulaires orientées pour des variétés abéliennes de dimension plus grande que 1. Nous utilisons pour cela l'espace des modules des variétés abéliennes munies d'un marquage ainsi que le système de coordonnées sur cet espace de module donné par les thêta constantes.

1.4 Tests d'aléa et théorie de l'information

Les générateurs d'aléa sont d'une importance capitale en cryptographie puisqu'ils sont utilisés afin de diversifier le comportement des algorithmes, pour générer des clefs, des marquants ou des vecteurs d'initialisation. Ainsi, les générateurs d'aléa constituent une brique essentielle et particulièrement critique d'un système implémentant des protocoles cryptographiques. Il est donc important de pouvoir évaluer la qualité du comportement aléatoire d'un générateur afin d'éviter tout problème pouvant amoindrir la sécurité. À cette fin, il est courant d'effectuer des tests statistiques sur un dispositif cryptographique afin de vérifier que la sortie se comporte en un certain sens comme une suite aléatoire. Cela peut être fait dans des circonstances variées : afin de valider un dispositif pendant la phase de conception, pour qualifier un composant durant sa production ou pour détecter des pannes en fonctionnement.

Ce sujet est relié à une production scientifique importante portant sur la définition d'une suite de tests [88], [102], sur la définition théorique de ce qu'est une suite aléatoire [59], [89], [56] et plus récemment sur la définition de tests associés à un modèle statistique général [90], [26].

D'une manière générale, afin de définir un test statistique, il est nécessaire de clarifier plusieurs choses :

– Quel caractère aléatoire est impliqué par le test. Cela veut dire donner une définition appropriée de ce qu'est une suite aléatoire et expliquer quel sur-ensemble de l'ensemble des suites aléatoires passera le test.
– Le lien entre la propriété d'aléa testée et le dispositif que l'on étudie. À cette fin, il est nécessaire d'expliciter un modèle mathématique pour le dispositif et d'en discuter la validité.
– Définir une fonction test qui répond par `oui` ou `non` étant donné une suite de sortie finie du dispositif.

Afin de définir une fonction test, il est d'usage de faire appel à la théorie des tests d'hypothèse (voir par exemple [102] ou bien [88]). L'objet de cette théorie [91] est de décider si un échantillon a été tiré selon une certaine distribution dans un modèle statistique, un modèle statistique étant une famille de distributions dépendant de certains paramètres. Afin de pouvoir utiliser cette théorie, il est nécessaire au préalable de donner une définition précise de ce qu'est un modèle statistique attaché à un test. Malheureusement, dans la littérature relative aux tests statistiques et à la théorie des tests d'hypothèse ce problème important est souvent ignoré. Par exemple, dans [102], le modèle statistique associé aux tests statistiques présentés ne sont pas toujours clairement définis. D'autres auteurs décrivent des modèles statistiques généraux associés à un test [90] ou bien une suite de tests dont la justification repose sur une définition empirique de ce qu'est une suite aléatoire [56]. Dans l'article [82], nous donnons une définition précise et générale de ce qu'est un test statistique qui englobe comme cas particuliers tous les tests usuels et nous expliquons comment attacher un modèle statistique particulier à chaque test. Ensuite, nous donnons une classification de tous les modèles statistiques tenant compte de leur capacité à distinguer certains types de déviation par rapport à un comportement vraiment aléatoire. Une retombée de cette approche générale qui utilise la théorie de l'information est qu'elle produit un formalisme utile pour comparer les différents modèles statistiques. Par exemple, il se peut que l'on veuille savoir si deux tests sont indépendants, i.e. si ils donnent des informations statistiques très différentes ou bien si au contraire un test est plus général qu'un autre. Certains résultats de l'article [82] permettent de répondre à ce genre de question. D'un point de vue plus théorique, nous déduisons de cette classification des analogues de certains résultats bien connus sur les suites équidistribuées. Nous dérivons aussi d'intéressantes conséquences pratiques.

Chapitre 2

Protocoles cryptographiques et logarithme discret

La théorie des preuves de protocole est l'une des branches les plus importantes et les plus originales de la cryptographie. Elle consiste en un ensemble de techniques permettant de modéliser les propriétés de sécurité d'un protocole et de prouver ces propriétés sous des hypothèses couramment admises relatives à la difficulté de certains problèmes calculatoires ou décisionnels. Deux objectifs principaux de cette théorie sont de proposer des modèles de primitives cryptographiques de plus en plus réalistes et d'autre part de prouver dans ces modèles des protocoles apportant des services de plus en plus évolués de la manière la plus efficace possible.

Comme exemple de modèles de primitives parmi les plus utilisés, on peut citer les fonctions à sens unique, les fonctions trappe, les familles de fonctions pseudo-aléatoires, le modèle de l'oracle aléatoire. Ces modèles sont plus ou moins réalistes : par exemple, alors que l'on garde espoir de pouvoir montrer un jour que les familles de fonctions utilisées pour faire du chiffrement sont des fonctions à sens unique, il est bien évident qu'une fonction de hachage ne se comportera jamais exactement comme le modèle de l'oracle d'aléa qui tire au hasard un motif binaire à chaque requête fraîche. Un autre modèle “approximatif”, qui d'une certaine manière sert à pallier à notre méconnaissance de la complexité des problèmes liés au logarithme discret, est le modèle du groupe générique. Le modèle du groupe générique ressemble en un certain sens à l'oracle d'aléa puisque dans la présentation originale de Shoup [115], l'oracle de groupe répond aussi de manière aléatoire à des requêtes fraîches. Dans la section 2.1, nous décrivons un modèle du groupe générique avec couplage donnant à l'adversaire le maximum de libertés possibles et nous donnons une généralisation de ce modèle pour des lois de groupes biaisées.

Dans la section 2.2, nous utilisons le modèle du groupe générique avec couplage afin de prouver la sécurité d'un protocole de diffusion sécurisé particulièrement efficace.

2.1 Le groupe générique avec couplage

L'objectif de cette section est de décrire les principaux résultats de [84] écrit en commun avec Thomas Sirvent.

2.1.1 Les groupes cycliques et leur représentation

2.1.1.1 Un unique groupe cyclique d'ordre n

Les groupes cycliques sont classifiés à isomorphisme près par leur ordre, i.e. deux groupes cycliques sont isomorphes si et seulement si ils ont même cardinal. En particulier, un groupe cyclique G est d'ordre $n \in \mathbb{N}^*$ si et seulement si il est isomorphe au groupe additif $\mathbb{Z}/n\mathbb{Z}$. Bien sûr, il peut exister plusieurs isomorphismes entre G et $\mathbb{Z}/n\mathbb{Z}$, mais nous pouvons considérer l'unique isomorphisme l de G engendré par g dans $\mathbb{Z}/n\mathbb{Z}$ tel que $l(g) = 1$. Nous remarquons que résoudre le problème du logarithme discret revient à calculer l'isomorphisme l entre deux groupes cycliques. Le problème du logarithme discret ne peut donc se comprendre si l'on raisonne à isomorphisme près comme cela est courant en mathématiques. Il faut plutôt s'intéresser à des familles de représentations de groupes cycliques qui sont données par des applications bijectives entre le modèle du groupe cyclique à n éléments, $\mathbb{Z}/n\mathbb{Z}$ et un ensemble de labels. L'idée du modèle du groupe générique est alors de considérer la plus grosse famille de représentations à savoir celle engendrée par toutes les applications bijectives de $\mathbb{Z}/n\mathbb{Z}$ sur les labels. L'objet de cette section est de donner des définitions précises pour de telles familles de représentations.

2.1.1.2 Structure de groupe héritée d'une bijection

Soit A un ensemble de n éléments. Toute application bijective f, de $\mathbb{Z}/n\mathbb{Z}$ dans A, munit A d'une structure de groupe, la loi de groupe $(+_f)$ et la loi d'inversion $(-_f)$ étant données par :

$$\forall (x,y) \in A^2,\ x +_f y = f(f^{-1}(x) + f^{-1}(y)) \ \text{ et } \ -_f x = f(-f^{-1}(x)). \tag{2.1}$$

Nous notons A_f l'ensemble A avec la structure de groupe ainsi déduite de f. Les éléments $f(0)$ et $f(1)$ sont respectivement l'élément neutre et un générateur distingué de A_f.

2.1.1.3 Famille générique de groupes cycliques

Soit $\mathbb{F}(A)$ l'ensemble de toutes les applications bijectives f, de $\mathbb{Z}/n\mathbb{Z}$ dans A. Pour tout sous-ensemble $S \subset \mathbb{F}(A)$, il est possible de considérer la famille $A_S = \{A_f, f \in S\}$. Cette famille est un ensemble de représentations du même groupe $\mathbb{Z}/n\mathbb{Z}$ sur l'ensemble A. Par la suite, nous utilisons la famille générique de groupes cycliques correspondant à la définition de [115] et donnée par la

Définition 1 (Famille générique de groupes cycliques)**.** Soit $B(n)$ l'ensemble des représentations binaires des entiers de $\{0, \ldots, n-1\}$. La famille $B(n)_{\mathbb{F}(B(n))}$ est appelée famille générique de groupes cycliques d'ordre n. La réunion sur $n \in \mathbb{N}^*$ des familles génériques de groupes cycliques d'ordre n est appelée la famille générique des groupes cycliques.

2.1.1.4 Représentation de groupes cycliques

Définition 2 (Famille de représentations de groupes)**.** Soit L un langage sur $\{0, 1\}$, i.e. un sous-ensemble de $\{0, 1\}^*$. Une famille de représentations de groupes cycliques sur le langage L est la donnée de :

- un ensemble de paramètres représentés par un ensemble infini dénombrable de suites binaires Ω ainsi qu'une fonction $c : \Omega \to \mathbb{N}^*$ calculable en temps polynomial, telle que $\forall N \in \mathbb{N}, \exists\, \alpha \in \Omega \,/\, c(\alpha) \geqslant N$,
- pour tout $\alpha \in \Omega$, un sous-ensemble fini L_α de L, de taille $c(\alpha)$, avec deux lois, $+_\alpha$ et $-_\alpha$, calculables en temps polynomial et deux éléments dans L_α : 0_α et g_α. Nous demandons à ce que $(L_\alpha, +_\alpha)$ soit un groupe cyclique d'ordre $c(\alpha)$ muni des lois de groupe $+_\alpha$ et d'inversion $-_\alpha$, et tel que 0_α et g_α soient respectivement l'élément neutre et un générateur du groupe. Nous supposons de plus que $\max\{|x|, x \in L_\alpha\} = O(\log_2(c(\alpha)))$.

2.1.1.5 Groupes cycliques avec couplage

Soit G un groupe cyclique d'ordre n. Un tel groupe est muni d'une structure canonique de $\mathbb{Z}$-module donnée par : $\forall (k, x) \in \mathbb{N} \times G, k.x = x + x + \ldots + x$ et $(-k).x = -(k.x)$. Nous notons $\hat{G}$ le dual de G, i.e. le groupe des $(\mathbb{Z}/n\mathbb{Z})$-formes linéaires sur G. Un couplage est une application $(\mathbb{Z}/n\mathbb{Z})$-bilinéaire d'un couple de groupes cycliques dans un groupe cyclique, où n est l'ordre commun de ces trois groupes. Il existe un couplage de $G \times \hat{G}$ dans $\mathbb{Z}/n\mathbb{Z}$, appelé le couplage canonique sur G défini par : $\forall (x, v) \in G \times \hat{G}$, $e_c(x, v) = v(x)$.

Soient G, G' et G'' trois groupes cycliques d'ordre n. Un couplage $e : G \times G' \to G''$ est dit parfait s'il est isomorphe comme couplage au couplage canonique de G, i.e. il existe trois isomorphismes $m : G \to G$, $\;m' : G' \to \hat{G}$ et $m'' : \mathbb{Z}/n\mathbb{Z} \to G''$ tels que $\forall (x, y) \in G \times G'$, $e(x, y) = m''(e_c(m(x), m'(y)))$. On a ainsi le diagramme suivant :

$$\begin{array}{ccc} G \times G' & \xrightarrow{e} & G'' \\ {\scriptstyle m}\downarrow \quad \downarrow{\scriptstyle m'} & & \uparrow{\scriptstyle m''} \\ G \times \hat{G} & \xrightarrow[e_c]{} & \mathbb{Z}/n\mathbb{Z} \end{array} .$$

Soit A un ensemble de n éléments. Soit $(f, h) \in \mathbb{F}(A)^2$, nous voudrions décrire tous les couplages parfaits possibles de $A_f \times A_f$ dans A_h. Par définition, un tel couplage se déduit de trois isomorphismes : m, m' et m'' tels que

$$\begin{array}{ccc} A_f \times A_f & \xrightarrow{e} & A_h \\ {\scriptstyle m}\downarrow \quad \downarrow{\scriptstyle m'} & & \uparrow{\scriptstyle m''} \\ A_f \times \widehat{A_f} & \xrightarrow[e_c]{} & \mathbb{Z}/n\mathbb{Z} \end{array} .$$

Sans perdre de généralité, on peut fixer $m = id,\ \ m'' = h$ et $m' = \hat{f}^{-1} \circ i \circ f^{-1}$, où i est n'importe quel isomorphisme de $\mathbb{Z}/n\mathbb{Z}$ dans $\widehat{\mathbb{Z}/n\mathbb{Z}}$, et où $\hat{f}$ est l'isomorphisme dual de f, i.e. $\hat{f} : v \in \widehat{G'} \mapsto v \circ f \in \hat{G}$. La donnée d'un couplage parfait est donc équivalent à celle d'un isomorphisme i.

Remarque 1. Le couplage parfait e est déterminé de manière unique par la valeur de $e(f(1), f(1))$. L'isomorphisme i est donc déterminé de même quand $e(f(1), f(1))$ est fixé. Donc, pour tout $(f, h) \in \mathbb{F}(A)^2$, il existe un unique couplage parfait e tel que $e(f(1), f(1)) = h(1)$.

Dans la suite, nous considérons seulement des groupes cycliques ayant un couplage parfait e tel que $e(f(1), f(1)) = h(1)$. Le couplage défini par $(x, y) \in {A_f}^2 \mapsto h(f^{-1}(x) \,.\, f^{-1}(y)) \in A_h$ vérifie cette propriété. Puisque c'est un couplage parfait, c'est l'unique couplage parfait mentionné dans la remarque précédente.

2.1.1.6 Famille de groupes cycliques avec couplage

Comme dans la section 2.1.1.3, nous considérons un ensemble A de n éléments et désignons par $\mathbb{F}(A)$ l'ensemble de toutes les applications bijectives de $\mathbb{Z}/n\mathbb{Z}$ dans A. Pour tout sous-ensemble $S \subset \mathbb{F}(A)$, soit $\mathcal{P}(A, S)$ une famille de deux représentations de groupes et un couplage paramétrisé par l'ensemble $\{(f, h) \in S^2\}$. D'une paire, $(f, h) \in S^2$, nous déduisons les groupes A_f et A_h d'après l'équation (2.1). De plus, nous considérons le couplage parfait e de $A_f \times A_f$ dans A_h défini par $\forall (x, y) \in (A_f)^2$, $e(x, y) = h\big(f^{-1}(x) \,.\, f^{-1}(y)\big)$.

Définition 3 (Famille générique de groupes avec couplage)**.** Soit $B(n)$ l'ensemble des représentations binaires des entiers dans $\{0, \ldots, n-1\}$. La famille $\mathcal{P}(B(n), \mathbb{F}(B(n)))$ est appelée famille générique de groupes cycliques d'ordre n avec couplage. L'union sur $n \in \mathbb{N}^*$ des familles génériques de groupes cycliques d'ordre n avec couplage est appelée la famille générique des groupes cycliques avec couplage.

Comme précédemment, les morphismes f et h peuvent se calculer à partir de la loi de groupe dans ${B(n)}_f$ et ${B(n)}_h$. Si le problème du logarithme discret est supposé difficile dans ces groupes, les applications inverses f^{-1} et h^{-1} ne sont pas facilement calculables.

2.1.1.7 Représentations de groupe cycliques avec couplage

Définition 4. Soient L et M deux langages sur $\{0, 1\}$. Une famille de représentation de groupes cycliques avec couplage sur les langages L et M est la donnée de deux familles de représentations de groupes cycliques $(\Gamma, (L_\gamma)_{\gamma \in \Gamma})$ sur L et $(\Delta, (M_\delta)_{\delta \in \Delta})$ sur M, un espace de paramètres $\Omega \subset \Gamma \times \Delta$, et pour tout $\alpha = (\gamma, \delta) \in \Omega$, un couplage parfait e_α de $L_\gamma \times L_\gamma$ dans M_δ, calculable en temps polynomial, tel que $e_\alpha(g_\gamma, g_\gamma) = g_\delta$.

Si Ω_1 (respectivement Ω_2) désigne l'ensemble des membres de gauche (respectivement de droite) d'éléments de Ω une telle famille est notée : $(\Omega, (L_\gamma)_{\gamma \in \Omega_1}$, $(M_\delta)_{\delta \in \Omega_2}, (e_\alpha)_{\alpha \in \Omega})$.

2.1.1.8 Problèmes Diffie-Hellman bilinéaires

Nous pouvons maintenant expliciter le problème Diffie-Hellman bilinéaire à l'aide de ce formalisme :

Définition 5. Soit $(\Omega, (L_\gamma)_{\gamma\in\Omega_1}, (M_\delta)_{\delta\in\Omega_2}, (e_\alpha)_{\alpha\in\Omega})$ une famille de représentation de groupes cycliques avec couplage sur les langages L et M. Dans cette famille

- un algorithme résolvant le problème bilinéaire Diffie-Hellman calcule l'élément $\log_{g_\gamma}(w)\,.\,e_{(\gamma,\delta)}(x,y)$ dans le groupe M_δ à partir des données $(\gamma,\delta)\in\Omega$, $(w,x,y)\in(L_\gamma)^3$;
- un algorithme résolvant le problème bilinéaire Diffie-Hellman décisionnel décide si z est $\log_{g_\gamma}(w)\,.\,e_{(\gamma,\delta)}(x,y)$ dans le groupe M_δ à partir des données $(\gamma,\delta)\in\Omega$, $(w,x,y)\in(L_\gamma)^3$, $z\in M_\delta$.

Comme précédemment, un algorithme résolvant un de ces problèmes dans la famille générique des groupes cycliques avec couplage a les mêmes entrées et sorties. Par contre, il n'a accès aux lois de groupes que par l'intermédiaire d'un oracle.

À partir de maintenant, afin d'exprimer les complexités en temps et espace, nous choisissons d'utiliser le modèle des machines de Turing avec oracle comme dans [104, pp. 36]. Nous définissons le coût d'un appel à un oracle comme une unité de temps.

Nous remarquons qu'il existe une réduction polynomiale évidente du problème bilinéaire Diffie-Hellman sur le problème du logarithme discret dans L_γ grâce à l'algorithme d'exponentiation rapide. En conséquence, en utilisant l'algorithme donné par Pollard dans [108] et la méthode proposée par Pohlig et Hellman [107], le problème bilinéaire Diffie-Hellman peut-être résolu avec une complexité en temps de $O(\sqrt{p})$ où p est le plus grand diviseur premier de $|L_\gamma|$. La partie suivante explique qu'il n'existe pas de meilleur algorithme pour résoudre le problème bilinéaire Diffie-Hellman dans la famille générique des représentations de groupes cycliques avec couplage.

2.1.2 Analyse de complexité

2.1.2.1 Une méthode générale

Dans cette section, nous présentons une méthode générale permettant d'évaluer la difficulté d'un problème dans la famille générique des groupes cycliques avec couplage. Un algorithme $\mathcal{A}$ résolvant un problème dans cette famille dispose au début des entrées suivantes : $n\in\mathbb{N}^*$, $(0_f, g_f, 0_h, g_h)\in(B(n))^4$, un r_0-uplet $(x_1,\ldots,x_{r_0})\in(B(n))^{r_0}$ et un s_0-uplet $(y_1,\ldots,y_{s_0})\in(B(n))^{s_0}$, où r_0+s_0 est le nombre de paramètres du problème donné et $(x_1,\ldots,x_{r_0},y_1,\ldots,y_{s_0})$ définit l'instance du problème.

De plus, $\mathcal{A}$ a accès à des oracles pour calculer les lois de groupe $+_f$ et $+_h$, les lois inverse $-_f$ et $-_h$, ainsi que le couplage e. Ces oracles sont bâtis en utilisant les bijections f et h choisies aléatoirement dans $\mathbb{F}(B(n))$, mais non dévoilées à $\mathcal{A}$. Les bijections f et h doivent être compatibles avec $(0_f, g_f, 0_h, g_h)$: $f(0)=0_f$, $f(1)=g_f$, $g(0)=0_h$, $g(1)=g_h$.

Nous supposons que l'algorithme $\mathcal{A}$ est une machine de Turing probabiliste. Nous évaluons sont temps de calcul par le nombre de ses appels aux oracles de groupe et de couplage. Nous souhaitons mesurer le comportement asymptotique de la probabilité de succès moyenne de $\mathcal{A}$ quand n tend vers l'infini, cette probabilité étant prise pour un n fixé sur l'ensemble des paires $(f, h) \in \mathbb{F}(B(n))^2$.

Afin d'analyser l'algorithme $\mathcal{A}$, nous maintenons deux séries de listes, R et S, à valeur dans $B(n) \times (\mathbb{Z}/n\mathbb{Z})(X_1, \ldots, X_n, Y_1, \ldots, Y_n)$ où $(\mathbb{Z}/n\mathbb{Z})(X_1, \ldots, Y_n)$ est le corps des fractions rationnelles en les variables $X_1, \ldots, X_n, Y_1, \ldots, Y_n$. Les listes R_k et S_k représentent la “connaissance” de $\mathcal{A}$ après k requêtes aux oracles. Les entiers r_k et s_k indiquent le nombre de variables utilisées dans les listes R_k et S_k. Les variables ρ_k et σ_k contiennent les cardinalités de R_k et S_k. Quand $\mathcal{A}$ fait un nouvel appel à un oracle, r_{k+1}, s_{k+1}, ρ_{k+1}, σ_{k+1}, R_{k+1} et S_{k+1} sont initialisées avec les valeurs de r_k, s_k, ρ_k, σ_k, R_k et S_k et mise à jour comme suit :

- dans le cas d'un appel $a +_f b$, $-_f a$ ou $e(a, b)$, si l'élément a (resp. b) n'est pas un second membre d'une paire de R_k, nous incrémentons de un r_{k+1} et ρ_{k+1}, posons $x_{r_{k+1}} = a$ (resp. b) et ajoutons $(x_{r_{k+1}}, X_{r_{k+1}})$ à R_{k+1},
- dans le cas d'un appel $a +_h b$ ou $-_h a$, si l'élément a (resp. b) n'est pas un second membre d'une paire dans S_k, nous incrémentons de un s_{k+1} et σ_{k+1}, nous posons $y_{s_{k+1}} = a$ (resp. b) et ajoutons $(y_{s_{k+1}}, Y_{s_{k+1}})$ à S_{k+1},
- si c est une réponse fraîche à une requête $a +_f b$ (resp. $-_f a$), la paire $(c, P_a + P_b)$ (resp. la paire $(c, -P_a)$) est ajoutée dans R_{k+1}, avec (a, P_a) et (b, P_b) dans R_k, et nous incrémentons de un ρ_{k+1},
- si c est une répondre fraîche à une requête $a +_h b$ (resp. $-_h a$), la paire $(c, P_a + P_b)$ (resp. la paire $(c, -P_a)$) est ajoutée dans S_{k+1}, avec (a, P_a) et (b, P_b) dans S_k, et nous incrémentons de un σ_{k+1},
- si c est une réponse fraîche à une requête $e(a, b)$, la paire $(c, P_a.P_b)$ est ajoutée dans S_{k+1}, avec (a, P_a) et (b, P_b) dans R_k, et nous incrémentons de un σ_{k+1}.

Nous remarquons que les règles précédentes impliquent que $\rho_{k+1} + \sigma_{k+1} \leqslant \rho_k + \sigma_k + 3$.

Définition 6 (Compatibilité)**.** Une paire de bijections $(f, h) \in \mathbb{F}(B(n))^2$ est dite compatible avec $(\mathsf{R}_k, \mathsf{S}_k)$ si :

- $\forall (v_R, P_R) \in \mathsf{R}_k$, $P_R(f^{-1}(x_1), \ldots, f^{-1}(x_{r_k}), h^{-1}(y_1), \ldots, h^{-1}(y_{s_k}))$ est défini et égal à $f^{-1}(v_R)$,
- $\forall (v_S, P_S) \in \mathsf{S}_k$, $P_S(f^{-1}(x_1), \ldots, f^{-1}(x_{r_k}), h^{-1}(y_1), \ldots, h^{-1}(y_{s_k}))$ est défini et égal à $h^{-1}(v_S)$.

Nous considérons un algorithme $\mathcal{A}$ après k appels aux oracles. D'après la définition 6, les oracles de groupe et de couplage initialisés avec n'importe quelle paire $(f, h) \in \mathbb{F}(B(n))^2$ compatible avec $(\mathsf{R}_k, \mathsf{S}_k)$ aurait produit les mêmes réponses aux appels de l'algorithme $\mathcal{A}$. L'idée générale derrière la définition 6 est qu'un algorithme $\mathcal{A}$, après k appels aux oracles, ne dispose d'aucun moyen de distinguer le problème défini par deux paires de bijections distinctes compatibles avec $(\mathsf{R}_k, \mathsf{S}_k)$.

Afin d'énoncer et de prouver notre lemme principal, nous avons besoin de définir les notions de collision et de cohérence :

Définition 7 (Collision)**.** Nous disons qu'il y a une collision dans $(\mathsf{R}_k, \mathsf{S}_k)$ s'il existe une paire $((v_1, P_1), (v_2, P_2))$ dans $(\mathsf{R}_k)^2$ ou dans $(\mathsf{S}_k)^2$ telle que : $v_1 = v_2$ et $P_1 \neq P_2$.

Définition 8 (Cohérence)**.** Un $(r+s)$-uplet $(\alpha_1,\ldots,\alpha_r,\beta_1,\ldots,\beta_s) \in (\mathbb{Z}/n\mathbb{Z})^{r+s}$ est dit cohérent avec un ensemble Π de fractions rationnelles de $(\mathbb{Z}/n\mathbb{Z})(X_1,\ldots,X_r,Y_1,\ldots,Y_s)$ si : $\forall P \in \Pi$, $P(\alpha_1,\ldots,\alpha_r,\beta_1,\ldots,\beta_s)$ est défini et $\forall (P_1,P_2) \in \Pi^2$, $\big(P_1 \neq P_2 \Longrightarrow P_1(\alpha_1,\ldots,\alpha_r,\beta_1,\ldots,\beta_s) \neq P_2(\alpha_1,\ldots,\alpha_r,\beta_1,\ldots,\beta_s)\big)$.

Lemme 1. Supposons que $n = p^\lambda$ pour p un nombre premier. Pour un k fixé, nous considérons une paire de listes $(\mathsf{R}_k,\mathsf{S}_k)$ sans collision, et un certain $(k+1)^{\text{ieme}}$ appel à un oracle. En écrivant tous les éléments de R_k et S_k sous leur forme réduite, soit d_1 le degré maximal des numérateurs de R_k et S_k, d_2 le degré maximal des dénominateurs de R_k et S_k, et soit $d = d_1 + d_2$. Si $\rho_k + \sigma_k < \sqrt{2p/3d}$, la probabilité prise sur toutes les paires de bijections compatibles avec $(\mathsf{R}_k,\mathsf{S}_k)$ qu'un nouvel appel produise une paire de listes $(\mathsf{R}_{k+1},\mathsf{S}_{k+1})$ avec collision est majorée par

$$\frac{6d\,(\rho_k+\sigma_k)}{2p - 3d\,(\rho_k+\sigma_k)^2}.$$

Pour une preuve de ce lemme nous référons à [84].

Grâce au lemme 1, nous disposons d'un moyen pour borner la probabilité d'occurrence d'une requête aux oracles provoquant une collision. De plus, il est facile d'interpréter le fait que l'attaquant trouve la bonne réponse au problème comme une requête virtuelle créant une collision. En utilisant deux fois le lemme précédent, il est alors facile de prouver des résultats de complexité dans le modèle du groupe générique. Afin d'illustrer ce mécanisme de preuve, nous nous intéressons maintenant à l'exemple du problème bilinéaire Diffie-Hellman sur la famille générique des groupes cycliques avec couplage.

Dans les deux corollaires suivants, nous supposons que n est une puissance d'un nombre premier : $n = p^\lambda$. Le cas plus général d'un ordre composé est traité dans [84]. Soit $\mathcal{A}$ un algorithme résolvant le problème Diffie-Hellman bilinéaire. Ses entrés sont $n \in \mathbb{N}^*$, ainsi que $(0_f, g_f, 0_h, g_h) \in B(n)^4$ et $(x_1,x_2,x_3) \in B(n)^3$. Au début, nous posons $\mathsf{R}_0 = \{(0_f,0),(g_f,1),(x_1,X_1),(x_2,X_2),(x_3,X_3)\}$ et $\mathsf{S}_0 = \{(0_h,0),(g_h,1)\}$.

Corollaire 1. Pour un $n = p^\lambda$ fixé, soit $k \leqslant (\sqrt{p/3} - 5)/3$. Si R_k et S_k contiennent des polynômes de degré au plus égal à 2, la probabilité que la paire de listes $(\mathsf{R}_k,\mathsf{S}_k)$ soit avec collision après k appels aux oracles est bornée par $2\,(3k+4)^2/(p - 3\,(3k+4)^2)$, la probabilité étant prise sur toutes les paires de bijections compatibles avec $(\mathsf{R}_k,\mathsf{S}_k)$.

Corollaire 2. Soit $\mathcal{A}$ un algorithme résolvant le problème bilinéaire Diffie-Hellman dans la famille générique des groupes cycliques avec couplage. Si $k < (\sqrt{2p/9} - 7)/3$, la probabilité de succès de $\mathcal{A}$ après k appels aux oracles, sur des groupes de tailles p^λ, quand $(\mathsf{R}_k,\mathsf{S}_k)$ est sans collision, est majorée par $18\,(3k+7)/(2p - 9\,(3k+7)^2)$.

Pour une preuve de ces résultats, nous renvoyons le lecteur à [84]. Des deux corollaires précédents, nous déduisons immédiatement le théorème :

Théorème 1. Soit $\mathcal{A}$ un algorithme résolvant le problème bilinéaire Diffie-Hellman dans la famille générique de groupes cycliques avec couplage. Nous ne faisons aucune hypothèse sur les capacités de calcul de l'attaquant $\mathcal{A}$. Après k appels aux oracles, sur un groupe d'ordre p^λ avec p premier si $k < (\sqrt{2p/9} - 7)/3$, la probabilité de succès de $\mathcal{A}$ est majorée par :

$$\frac{2\,(3k+4)^2}{p - 3\,(3k+4)^2} + \frac{18\,(3k+7)}{2p - 9\,(3k+7)^2}.$$

2.1.3 Famille pseudo-aléatoire de groupes cycliques

Dans cette section, nous introduisons la notion de famille pseudo-aléatoire de groupes cycliques. Une famille pseudo-aléatoire de groupes est la même chose qu'une famille générique de groupes excepté que la loi de groupe est tirée non pas de manière aléatoire dans l'ensemble de toutes les lois de groupes : la loi de groupe suit une distribution spécifique dont on suppose qu'elle est calculatoirement indistinguable de la distribution uniforme. Nous construisons une famille pseudo-aléatoire de groupes cycliques à partir d'une famille de permutations fortement pseudo-aléatoires.

Soit $\mathfrak{P}$ un ensemble de permutations sur $B(n)$. La notation $f \leftarrow \mathfrak{P}$ signifie que f est tirée selon la distribution uniforme dans $\mathfrak{P}$. Un distingueur $\mathcal{D}$ est une machine de Turing qui a accès à des permutations sur $B(n)$ par l'intermédiaire d'oracles et produit un unique bit. Dans le contexte de l'indistingabilité forte entre deux familles de permutations $\mathfrak{P}_1$ et $\mathfrak{P}_2$ (voir [87] pour plus de détails), un distingueur $\mathcal{D}$ a accès à f et f^{-1}, avec $f \leftarrow \mathfrak{P}_1$ ou $f \leftarrow \mathfrak{P}_2$. Si $\mathcal{D}$ est exécuté pendant un temps t et fait q requêtes aux oracles, son avantage est défini par la formule suivante :

$$\mathrm{Adv}^{\text{s-prp}}_{\mathfrak{P}_1,\mathfrak{P}_2}(\mathcal{D}, t, q) = \left| \Pr_{f\leftarrow\mathfrak{P}_1}[{\mathcal{D}_{t,q}}^{f,f^{-1}} = 1] - \Pr_{f\leftarrow\mathfrak{P}_2}[{\mathcal{D}_{t,q}}^{f,f^{-1}} = 1] \right|.$$

Nous disons que $\mathfrak{P}_1$ et $\mathfrak{P}_2$ sont (ϵ, t, q)-fortement indistinguables si pour tout distingueur $\mathcal{D}$, l'avantage $\mathrm{Adv}^{\text{s-prp}}_{\mathfrak{P}_1,\mathfrak{P}_2}(\mathcal{D}, t, q)$ est majoré par ϵ. Nous disons que $\mathfrak{P}$ est une famille de permutations (ϵ, t, q)-fortement pseudo-aléatoires si elle est (ϵ, t, q)-fortement indistinguable de l'ensemble $\mathfrak{S}_n$ de toutes les permutations sur $B(n)$.

Définition 9. Soit $\mathfrak{P}$ une famille de permutations (ϵ, t, q)-pseudo-aléatoire sur $B(n)$. La famille pseudo-aléatoire de groupes cycliques associée à $\mathfrak{P}$ est l'ensemble des groupes définis par une permutation f dans $\mathfrak{P}$, avec l'élément neutre $f(0)$, le générateur $f(1)$ et les lois de groupe $+_f$ et $-_f$, définies comme dans la section 2.1.1.3.

Comme pour la famille générique de groupes cycliques, les lois de groupes sont données seulement par l'intermédiaire d'oracles. De cette manière, il est clair qu'une famille générique de groupes cycliques est pseudo-aléatoire. En conséquence de quoi, la notion de famille pseudo-aléatoire de groupes constitue une généralisation de la notion de famille générique de groupes cycliques.

Nous pouvons maintenant définir l'avantage d'un adversaire résolvant un problème de type logarithme discret dans une famille pseudo-aléatoire de groupes cycliques : soit $\mathcal{A}$ un adversaire ayant accès à des lois de groupes sur $B(n)$ par l'intermédiaire d'oracles $+_f$ et $-_f$. Si $\mathcal{A}$ s'exécute en temps t et fait q requêtes aux oracles, son avantage contre le problème du logarithme discret, le problème Diffie-Hellman et le problème décisionnel Diffie-Hellman sont respectivement définis par :

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{DL}}_{\mathfrak{P}}(\mathcal{A},t,q) &= \Pr_{f \leftarrow \mathfrak{P},\, x \in B(n)}[\mathcal{A}_{t,q}{}^{+_f,-_f}(f(0),f(1),x) = \log_{f(1)}(x)],\\
\mathrm{Adv}^{\mathrm{DH}}_{\mathfrak{P}}(\mathcal{A},t,q) &= \Pr_{f \leftarrow \mathfrak{P},\, (x,y) \in B(n)^2}[\mathcal{A}_{t,q}{}^{+_f,-_f}(f(0),f(1),x,y) = \log_{f(1)}(x)\,.\,y],\\
\mathrm{Adv}^{\mathrm{DDH}}_{\mathfrak{P}}(\mathcal{A},t,q) &= \Big| \Pr_{f \leftarrow \mathfrak{P},\, (x,y) \in B(n)^2}[\mathcal{A}_{t,q}{}^{+_f,-_f}(f(0),f(1),x,y,\log_{f(1)}(x)\,.\,y) = 1]\\
&\qquad - \Pr_{f \leftarrow \mathfrak{P},\, (x,y,z) \in B(n)^3}[\mathcal{A}_{t,q}{}^{+_f,-_f}(f(0),f(1),x,y,z) = 1]\Big|.
\end{aligned}$$

Les maximums de ces avantages pris sur tous les adversaires $\mathcal{A}$ sont notés respectivement $\mathrm{Adv}^{\mathrm{DL}}_{\mathfrak{P}}(t,q)$, $\mathrm{Adv}^{\mathrm{DH}}_{\mathfrak{P}}(t,q)$, et $\mathrm{Adv}^{\mathrm{DDH}}_{\mathfrak{P}}(t,q)$. Les théorèmes 2 et 3 donnent des bornes sur ces probabilités, quand $\mathfrak{P}$ est une famille (ϵ,t,q)-fortement pseudo-aléatoire de permutations sur $B(n)$.

Théorème 2. Soit $\mathfrak{P}$ une famille (ϵ,t,q)-fortement pseudo-aléatoire de permutations sur $B(n)$. Nous avons,

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{DL}}_{\mathfrak{P}}(t,q/3-1) &\leqslant \mathrm{Adv}^{\mathrm{DL}}_{\mathfrak{S}_n}(t,q/3-1) + \epsilon,\\
\mathrm{Adv}^{\mathrm{DH}}_{\mathfrak{P}}(t,q/3-2) &\leqslant \mathrm{Adv}^{\mathrm{DH}}_{\mathfrak{S}_n}(t,q/3-2) + \epsilon.
\end{aligned}$$

Théorème 3. Soit $\mathfrak{P}$ une famille (ϵ,t,q)-fortement pseudo-aléatoire de permutations sur $B(n)$. Alors

$$\mathrm{Adv}^{\mathrm{DDH}}_{\mathfrak{P}}(t,q/3-2) \leqslant \mathrm{Adv}^{\mathrm{DDH}}_{\mathfrak{S}_n}(t,q/3-2) + 2\epsilon.$$

Pour une preuve de ces théorèmes nous renvoyons le lecteur à [84].

2.1.4 Conclusion

Nous avons revisité la notion de groupe générique afin de décrire un modèle qui contient toutes le fonctionnalités déjà décrites dans la littérature : la possibilité de soumettre une requête ou bien fraîche ou bien correspondant à des fractions rationnelles des exposants mérite d'être mentionnée. Nous avons prouvé une borne pour le problème qui consiste à trouver une collision dans ce modèle (lemme 1). À partir de cette borne, il est facile de déduire des bornes précises pour tous les problèmes usuels de type logarithme discret : nous avons expliqué comment le lemme 1 peut être utilisé de manière systématique pour prouver des résultats de ce type. Nous renvoyons le lecteur à l'article [84] pour des exemples détaillés.

Il est possible de généraliser le modèle du groupe générique en introduisant la notion de groupe pseudo-aléatoire et de prouver une réduction de problèmes usuels dans le groupe pseudo-aléatoire à leur sécurité dans le modèle du groupe générique sous l'hypothèse permutation fortement pseudo-aléatoire. Comme conséquence de la réduction que nous prouvons, le modèle du groupe pseudo-aléatoire n'apporte pas de nouveauté sur la plan de la sécurité. Pourtant, il constitue une amélioration par rapport au modèle du groupe générique sur le plan du réalisme. Le modèle du groupe pseudo-aléatoire n'est cependant toujours pas satisfaisant. En effet, dans les familles de groupes utilisées en cryptographie, la famille de permutations sous-jacente aux différentes lois de groupes est loin d'être pseudo-aléatoire. Par exemple, si la famille de groupes est la famille des éléments multiplicatifs d'un corps fini, pour une cardinalité donnée n il y a seulement $\phi(n)$ permutations possibles où ϕ est la fonction indicatrice d'Euler. Il serait intéressant de généraliser la notion de groupe pseudo-aléatoire afin de la rendre encore plus réaliste.

2.2 Un protocole de diffusion basé sur les attributs

L'objet de cette section est d'exposer les principaux résultats de [83] écrit en commun avec Thomas Sirvent. Dans cet article, nous décrivons un schéma de diffusion particulièrement efficace dans certaines situations réalistes de type télévision à péage. Le problème est de diffuser des messages à un ensemble dit privilégié d'utilisateurs et de pouvoir changer cet ensemble utilisateurs privilégiés de la manière la moins couteuse possible en terme de bande passante et de calcul pour le récepteur.

2.2.1 Un schéma de diffusion basé sur les attributs

Dans cette section, nous donnons une définition formelle de ce qu'est un schéma de diffusion basé sur les attributs. Les définitions suivantes sont de simples adaptations de [10, 6] afin de tenir compte de groupes d'utilisateurs.

Dans les applications qui nous intéressent, l'objectif est de gérer un grand nombre d'utilisateurs, et un grand nombre de groupes d'utilisateurs (en pratique, nous avons besoin que chaque utilisateur soit associé à un groupe singleton). Chaque utilisateur appartient à un certain nombre de groupe d'utilisateurs.

Nous formalisons cela de la manière suivante. Soit $\mathcal{U}$ l'ensemble de tous les utilisateurs. Nous représentons un élément de $\mathcal{U}$ par un entier dans $\{1, \dots, n\}$. Un groupe d'utilisateurs est un sous-ensemble de $\mathcal{G}$ de $\mathcal{U}$. En prenant le point de vue inverse, pour un nombre fixé l de groupes d'utilisateurs, nous pouvons associer à un utilisateur $u \in \mathcal{U}$ l'ensemble des groupes auxquels il appartient : $\mathcal{B}(u) = \{i \in \{1, \dots, l\} \,/\, u \in \mathcal{G}_i\} \subset \{1, \dots, l\}$.

Un schéma de diffusion à clef publique basé sur les attributs ayant pour paramètre de sécurité λ est un triplet d'algorithmes probabilistes :

- ***Initialisation***$(\lambda, n, (\mathcal{B}(u))_{1 \leq u \leq n})$: prend en entrée le paramètre de sécurité λ, le nombre d'utilisateurs n, et les groupes d'utilisateurs. Il produit une clef de chiffrement EK et n clefs de déchiffrement $(\mathrm{dk}_u)_{1 \leq u \leq n}$.
- ***Chiffrement*** $(\mathrm{EK}, \mathcal{B}^N, \mathcal{B}^R)$: prend en entrée une clef de chiffrement EK et deux ensembles de groupes $\mathcal{B}^N$ et $\mathcal{B}^R$. Il produit un entête hdr et une clef de chiffrement de message $K \in \mathcal{K}$, où $\mathcal{K}$ est l'ensemble fini des clefs de chiffrement de message.
- ***Déchiffrement*** $(\mathrm{dk}_u, \mathrm{hdr})$: prend en entrée une clef de déchiffrement associée à un utilisateur u et un entête hdr. Si l'entête hdr provient d'un chiffrement

utilisant $(\mathcal{B}^N, \mathcal{B}^R)$ tel que $\mathcal{B}^N \subset \mathcal{B}(u)$ et $\mathcal{B}(u) \cap \mathcal{B}^R = \emptyset$, alors il produit la clef de déchiffrement du message $K \in \mathcal{K}$. Dans le cas contraire, il sort $\perp$.

Pendant le processus de chiffrement, un message M est chiffré avec une clef K et le chiffré résultant C est envoyé avec l'entête hdr. Les utilisateurs appartenant à tous les groupes mentionnés dans $\mathcal{B}^N$ (groupes requis) et en dehors de tous les groupes de $\mathcal{B}^R$ (groupes révoqués) peuvent calculer la clef K à partir de l'entête hdr et de leur clef de déchiffrement dk_u. En utilisant la clef K, un utilisateur peut retrouver M à partir de C.

Il convient de noter que dans ces définitions, la clef de déchiffrement et l'entête sont les seuls éléments dont un utilisateur a besoin pour le calcul de la clef K. La connaissance de la clef de chiffrement et de l'ensemble des utilisateurs privilégiés n'est pas nécessaire pour le déchiffrement. La taille de l'entête correspond alors exactement au coût en terme de bande passante pour le schéma de diffusion. En effet, dans notre schéma, la connaissance de l'ensemble des utilisateurs privilégiés est implicitement contenue dans l'entête, encodée par les attributs correspondant aux groupes requis et aux groupes révoqués.

Dans cette description, il semble à première vue que nous ne permettions pas un chiffrement pour un ensemble arbitraire d'utilisateurs privilégiés alors que c'est une propriété habituellement requise pour un schéma de diffusion. Un ensemble quelconque d'utilisateurs privilégié peut toutefois être représenté comme la réunion des groupes d'utilisateurs accessibles via un "chiffrement simple" (en fait, il suffit que chaque utilisateur appartienne à un groupe singleton). Différents chiffrements simples peuvent être utilisés pour chiffrer une clef commune. Le message complet peut alors être envoyé chiffré avec cette clef commune.

2.2.2 Un modèle de sécurité

Dans cette section, nous expliquons ce que nous entendons par la sécurité sémantique des schémas de diffusion basés sur les attributs. L'adversaire est supposé statique comme dans les modèles déjà connus : la seule différence avec les définitions standards est que les groupes d'utilisateurs sont donnés à l'adversaire avant le début du jeu auquel participent le challenger et l'adversaire $\mathcal{A}$:

- Le challenger et l'adversaire reçoivent un nombre l de groupes d'utilisateurs fixé, définis par $(\mathcal{B}(u))_{1 \le u \le n}$.
- L'adversaire $\mathcal{A}$ produit deux ensembles de groupes $\mathcal{B}^N$ et $\mathcal{B}^R$ correspondant à une configuration qu'il souhaite attaquer.
- Le challenger lance *Initialisation*$(\lambda, n, (\mathcal{B}(u))_{1 \le u \le n})$ et donne à $\mathcal{A}$ la clef de chiffrement EK et la clef de déchiffrement dk_u correspondant aux utilisateurs que l'adversaire pourrait contrôler, i.e. tels que $\mathcal{B}^N \cap \mathcal{B}(u) \neq \mathcal{B}^N$ ou $\mathcal{B}^R \cap \mathcal{B}(u) \neq \emptyset$.
- Le challenger lance *Chiffrement*$(\mathrm{EK}, \mathcal{B}^N, \mathcal{B}^R)$ et obtient un entête hdr et une clef $K \in \mathcal{K}$. Ensuite, le challenger tire un bit aléatoire b et pose $K_b = K$, choisit aléatoirement K_{1-b} dans $\mathcal{K}$ et envoie (hdr, K_0, K_1) à l'adversaire $\mathcal{A}$.
- L'adversaire $\mathcal{A}$ produit un bit b'.

L'adversaire $\mathcal{A}$ gagne le jeu précédent si $b' = b$. L'avantage de $\mathcal{A}$ dans ce jeu, paramètré par $(\lambda, n, (\mathcal{B}(u))_{1\leq u\leq n})$, est $|2\Pr[b' = b]-1|$, où les probabilités sont prises sur les choix de b et de tous les bits aléatoires utilisés pendant la simulation des algorithmes *Initialisation* et *Chiffrement*.

Un schéma de diffusion basé sur les attributs est dit sémantiquement sûr contre toute collusion statique maximale si pour tout adversaire $\mathcal{A}$ probabiliste polynomial en le paramètre λ et pour tout groupe d'utilisateurs $(\mathcal{B}(u))_{1\leq u\leq n}$ dont le nombre est majoré par l, $\mathrm{Adv}^{\mathrm{ind}}(\lambda, n, (\mathcal{B}(u)), \mathcal{A})$ est une fonction négligeable en λ quand n et l sont au plus polynomials en λ.

À partir d'un tel schéma sémantiquement sûr, il est possible de construire des schémas sûrs contre un modèle plus fort : l'utilisation d'une transformation générique, comme celle présentée dans [38, 39, 103] permet pour un coût négligeable d'obtenir un schéma sûr contre un attaquant à chiffré choisi dans le modèle de l'oracle aléatoire. Cela explique pourquoi notre modèle de sécurité se limite aux attaques à clairs choisis.

2.2.3 Des groupes d'utilisateurs bien choisis

Dans les applications de diffusion réelles, il existe bien souvent des groupes d'utilisateurs adaptés, parce que les utilisateurs sont classifiés par exemple par type ou période d'abonnement. Ces groupes sont faciles à gérer avec un schéma de diffusion basé sur les attributs en utilisant simplement un attribut adapté pour chaque groupe d'utilisateurs.

Dans certaines circonstances, il peut arriver que le groupe des utilisateurs privilégiés ne s'exprime pas facilement avec ces groupes d'utilisateurs adaptés. Même si cela arrive rarement, il est préférable de pouvoir gérer ce genre de situation.

Une solution consiste à ajouter des attributs supplémentaires à l'ensemble des attributs correspondant aux groupes adaptés. Ces nouveaux attributs décrivent un arbre binaire sur l'ensemble des utilisateurs et permet la même gestion des utilisateurs que dans un schéma "Subset-Layer". De manière plus précise, nous plaçons un utilisateur à chaque feuille d'un arbre binaire, chaque noeud correspondant à un nouvel attribut et chaque utilisateur recevant les attributs correspondant aux noeuds de ses parents. Au plus $2n$ nouveaux attributs sont ajoutés et un utilisateur appartient à au plus $\lceil \log_2(n) \rceil + 1$ nouveaux groupes.

Avec cette configuration, il existe un attribut pour chaque utilisateur et ce simple fait garantit que tout sous-ensemble d'utilisateurs peut être décrit avec des attributs. De plus, un chiffrement simple pour lequel l'ensemble des utilisateurs privilégiés correspond aux membres d'un unique groupe en excluant les membres d'un autre groupe donne au moins les mêmes familles de sous-ensembles que la méthode SD présentée dans [99]. L'efficacité de notre schéma de diffusion basé sur les attributs est donc au moins aussi bonne que la méthode SD dans le cas d'un ensemble arbitraire d'utilisateurs privilégiés.

2.2.4 Construction

Dans cette section, nous décrivons l'algorithme *Initialisation*, *Chiffrement* et *Déchiffrement* pour un schéma de diffusion à clef publique reposant sur la difficulté d'un problème dans une famille de groupes avec couplage. La correction du protocole peut ensuite être vérifiée.

2.2.4.1 L'algorithme d'initialisation

À partir du paramètre de sécurité λ, la première étape de l'initialisation consiste en la construction de $(\mathbb{G}_1, \mathbb{G}_2, g_1, g_2, e, p)$ où

- p est un nombre premier de longueur λ,
- $\mathbb{G}_1$ et $\mathbb{G}_2$ sont deux groupes cycliques d'ordre premier p,
- e est un accouplement non-dégénéré de $\mathbb{G}_1 \times \mathbb{G}_1$ dans $\mathbb{G}_2$,
- g_1 est un générateur de $\mathbb{G}_1$ et $g_2 = e(g_1, g_1)$.

Quatre éléments (α, β, γ et δ) sont choisis aléatoirement dans $(\mathbb{Z}/p\mathbb{Z})^*$. Chaque groupe d'utilisateurs $\mathcal{G}_i$ mentionné dans $(\mathcal{B}(u))_{1 \le u \le n}$ est alors associé à un attribut μ_i choisi aléatoirement dans $(\mathbb{Z}/p\mathbb{Z})$, de manière à ce que tous les attributs soient deux à deux distincts et différents de α. Un autre attribut μ_0 est choisi avec les mêmes contraintes. La clef de chiffrement est donnée par :

$$\mathrm{EK} = \Big(g_1\,,\, \beta\,\gamma\,\delta\, g_1\,,\, (\mu_i)_{0 \le i \le l}\,,\, \big(\alpha^i\, g_1\big)_{0 \le i \le l}\,,\, \big(\alpha^i\,\gamma\, g_1\big)_{0 \le i \le l}\,,\, \big(\alpha^i\,\delta\, g_1\big)_{0 \le i \le l}\Big)\,.$$

Pour chaque utilisateur $u \in \mathcal{U}$, s_u est choisi aléatoirement dans $(\mathbb{Z}/p\mathbb{Z})^*$. Soit $\Omega(u)$ l'ensemble des attributs correspondants aux groupes auxquels il appartient : $\Omega(u) = \{\mu_i \in (\mathbb{Z}/p\mathbb{Z}) \,/\, i \in \mathcal{B}(u)\}$. Soit $l(u)$ la taille de $\Omega(u)$, i.e. le nombre de groupes contenant u. Soit $\Pi(u) = \prod_{\mu \in \Omega(u)} (\alpha - \mu)$. La clef de déchiffrement de u est :

$$\mathrm{dk}_u = \Big(\Omega(u), (\beta + s_u)\,\delta\, g_1\,,\, \gamma\, s_u\, \Pi(u)\, g_1\,,\, \big(\alpha^i\,\gamma\,\delta\, s_u\, g_1\big)_{0 \le i < l(u)}\Big)\,.$$

2.2.4.2 L'algorithme de chiffrement

Si $\mathcal{B}^N \cap \mathcal{B}^R \neq \emptyset$, l'algorithme de chiffrement s'arrête et retourne $\perp$ puisqu'un utilisateur ne peut être simultanément à l'intérieur et à l'extérieur d'un groupe d'utilisateurs. Sinon, posons $\Omega^N = \{\mu_i \,/\, i \in \mathcal{B}^N\}$ et $\Omega^R = \{\mu_i \,/\, i \in \mathcal{B}^R\}$. Soit $l^N = |\mathcal{B}^N|$ le nombre de groupes requis et $l^R = |\mathcal{B}^R|$ le nombre de groupes révoqués[1]. Soient $\Pi^N = \prod_{\mu \in \Omega^N} (\alpha - \mu)$, $\Pi^R = \prod_{\mu \in \Omega^R} (\alpha - \mu)$ et $\Pi^{NR} = \Pi^N \Pi^R$. Soit z choisi aléatoirement dans $(\mathbb{Z}/p\mathbb{Z})^*$. Le résultat du chiffrement est :

$$\mathrm{hdr} = \Big(\Omega^N, \Omega^R, z\,\Pi^{NR}\, g_1\,,\, \gamma\, z\,\Pi^N\, g_1\,,\, \big(\alpha^i\,\delta\, z\, g_1\big)_{0 \le i < l^R}\Big)\,,\; K = \beta\,\gamma\,\delta\, z\,\Pi^N\, g_2.$$

Tous les éléments de *hdr* peuvent être calculés en utilisant seulement la clef de chiffrement EK.

[1]Il faut faire une légère modification dans le cas où $\mathcal{B}_R$ est vide : le chiffrement considère alors que le groupe virtuel ne contenant aucun utilisateur est révoqué et alors $\Omega^R = \{\mu_0\}$, $l^R = 1$.

2.2.4.3 Algorithme de déchiffrement

Nous considérons ici l'entête de déchiffrement hdr ayant une clef de déchiffrement dk_u.

$$\begin{cases} \mathrm{dk}_u = (\Omega(u), \mathrm{dk}_1, \mathrm{dk}_2, \mathrm{dk}_{3,0}, \ldots, \mathrm{dk}_{3,l(u)-1})\,, \\ \mathrm{hdr} = (\Omega^N, \Omega^R, \mathrm{hdr}_1, \mathrm{hdr}_2, \mathrm{hdr}_{3,0}, \ldots, \mathrm{hdr}_{3,l^R-1})\,. \end{cases}$$

Le recepteur u est valide pour cet entête si $\Omega(u)$ contient Ω^N et si l'intersection de Ω^R avec $\Omega(u)$ est vide. Afin de déchiffrer l'entête, un utilisateur valide u utilise un algorithme d'euclide étendu sur les polynômes $\prod_{\mu\in(\Omega^N\cup\Omega^R)}(X-\mu)$ et $\prod_{\mu\in\Omega(u)}(X-\mu)$. Il obtient deux polynômes unitaires $V(X) = \sum_{0\leq i<l(u)} v_i X^i$ et $W(X) = \sum_{0\leq i<l^R} w_i X^i$, dans $(\mathbb{Z}/p\mathbb{Z})[X]$ tels que :

$$V(X) \prod_{\mu\in(\Omega^N\cup\Omega^R)} (X-\mu) + W(X) \prod_{\mu\in\Omega(u)} (X-\mu) = \prod_{\mu\in\Omega^N} (X-\mu).$$

À partir de ces polynômes, le recepteur peut calculer la clef :

$$K(\mathrm{dk}_u, \mathrm{hdr}) = e(\mathrm{dk}_1, \mathrm{hdr}_2) - e\left(\sum_{i=0}^{l(u)-1} v_i\, \mathrm{dk}_{3,i}\,,\ \mathrm{hdr}_1\right) - e\left(\mathrm{dk}_2\,,\ \sum_{i=0}^{l^R-1} w_i\, \mathrm{hdr}_{3,i}\right).$$

2.2.4.4 Preuve de correction

Si dk_u est la clef de déchiffrement valide pour un utilisateur u, si hdr est un entête construit en utilisant l'algorithme de chiffrement et si u est un utilisateur valide pour hdr alors le déchiffrement donne :

$$K(\mathrm{dk}_u, \mathrm{hdr}) = (\beta + s_u)\, \gamma\, \delta\, z\, \Pi^N g_2 - \gamma\, \delta\, z\, s_u\, V(\alpha)\, \Pi^{NR} g_2 - \gamma\, \delta\, z\, s_u\, W(\alpha)\, \Pi(u)\, g_2.$$

Par définition des deux polynômes V et W, nous avons la relation $V(\alpha)\,\Pi^{NR} + W(\alpha)\Pi(u) = \Pi^N$. La clef calculée est alors exactement la clef associée à l'entête du déchiffré :

$$K(\mathrm{dk}_u, \mathrm{hdr}) = (\beta + s_u)\, \gamma\, \delta\, z\, \Pi^N g_2 - \gamma\, \delta\, z\, s_u\, \Pi^N g_2 = \beta\, \gamma\, \delta\, z\, \Pi^N g_2.$$

2.2.5 Sécurité du schéma

Le schéma précédent peut être prouvé de différentes manières. La stratégie habituelle consiste tout d'abord à définir des hypothèses de sécurité et à les prouver dans le modèle du groupe générique avec couplage. La réduction de la sécurité du schéma sur les hypothèses conclut alors la preuve. Suivant cette stratégie, nous avons besoin d'une nouvelle hypothèse de sécurité qui est une extension de la version décisionnelle du problème de l'Exposant Diffie-Hellman Général (GDHE en anglais), précisément étudiée dans la version longue de [9]. Pour plus de clarté, nous préférons donner dans [83] une preuve directe dans le modèle du groupe générique avec couplage. Cela donne le

Théorème 4. Dans le modèle du groupe générique avec couplage, l'avantage d'un adversaire pour le problème défini dans la partie 2.2.2 du schéma de diffusion basé sur les attributs présenté dans la section 2.2.4, produisant au plus q requêtes aux oracles est bornée par :

$$\frac{(l+2)\,(q+2nl+6n+6l+14)^2}{p-\sqrt{p}},$$

où n est le nombre d'utilisateurs et l le nombre de groupes d'utilisateurs.

2.2.6 Conclusion

Dans l'article [83] nous avons conçu un nouveau schéma de diffusion à clef publique particulièrement intéressant quand il s'agit de gérer des groupes d'utilisateurs définis par des conjonctions ou exclusions d'attributs. Nous avons donné une application pratique pour laquelle aucun des schémas de diffusion existant n'apportaient une solution satisfaisante.

Nous avons indiqué une manière générique d'utiliser les attributs afin de gérer efficacement les groupes d'utilisateurs. Enfin, nous avons prouvé que notre schéma est sémantiquement sûr contre une collusion statique maximale dans le modèle du groupe générique avec couplage.

Il serait intéressant de voir de quelle manière il est possible d'améliorer la structure d'accès de notre schéma en implémentant efficacement le OU, ou bien une fonctionnalité à seuil. Nous pensons aussi que le problème sous-jacent à notre schéma, fondé sur la reconstruction du plus grand diviseur commun de deux polynômes, pourrait avoir d'autres applications intéressantes.

Chapitre 3

Algorithmes de comptage de points

Comme nous l'avons expliqué dans l'introduction, un préalable important à l'utilisation des jacobiennes de courbes algébriques définies sur un corps fini est de pouvoir calculer la cardinalité du groupe formé par leurs points rationnels. En effet, il convient d'éliminer les courbes faibles parce que produisant des groupes d'ordre friable.

Les algorithmes de comptage de points rapides reposent sur le calcul du polynôme caractéristique du morphisme de Frobenius à partir duquel il est facile de retrouver le nombre de points rationnels de la jacobienne d'une courbe. De manière plus précise, considérons une courbe algébrique C projective lisse de genre g définie sur le corps fini $\mathbb{F}_q$ à $q = p^n$ éléments, p premier, et notons $J(C)$ sa jacobienne. Soit Σ le Frobenius à la puissance n agissant sur la clôture algébrique $\overline{\mathbb{F}}_q$ de $\mathbb{F}_q$. Nous rappelons que le morphisme de Frobenius F agissant sur $J(C)$ est le morphisme inséparable de degré q^g donné par l'action de Σ sur les coordonnées affines des points géométriques de $J(C)$, $(x_1, \ldots, x_d) \to (x_1^q, \ldots, x_d^q)$. Nous pouvons définir l'ensemble des points $\mathbb{F}_q$-rationnels de C comme l'ensemble des points géométriques de C invariants par l'action de F.

Pour ℓ un entier premier à p, nous notons $J[\ell]$ le sous-groupe des points de ℓ-torsion de $J(C)$. Ce sous-groupe est isomorphe à $(\mathbb{Z}/\ell\mathbb{Z})^{2g}$. De plus, si pour $i \in \mathbb{N}^*$, nous désignons par $[i]$ la multiplication par i dans J, alors pour $i > j$, nous disposons d'une projection canonique $p_{ij} : J[\ell^i] \to J[\ell^j]$ donnée par le morphisme $[\ell^{i-j}]$. La famille de projections p_{ij} vérifie les relations de compatibilité usuelles de manière à ce que nous puissions définir le module de Tate comme la limite projective $T_\ell = \varprojlim J[\ell^i]$. Le module de Tate T_ℓ est un $\mathbb{Z}_\ell$-module libre de rang $2g$. Les éléments du groupe de Galois $\mathrm{Gal}(\overline{\mathbb{F}}_q/\mathbb{F}_q)$ agissent sur $J(C)$ et en conséquence sur $J[\ell]$ de manière compatible avec les morphismes de projection p_{ij} et en passant à la limite projective, on obtient un morphisme $\rho_\ell : \mathrm{Gal}(\overline{\mathbb{F}_q}/\mathbb{F}_q) \to \mathrm{Aut}_{\mathbb{Z}_\ell}(T_\ell)$, qui n'est autre que la représentation ℓ-adique de $\mathrm{Gal}(\overline{\mathbb{F}_q}/\mathbb{F}_q)$.

Nous définissons le polynôme caractéristique χ_F du morphisme de Frobenius à la puissance n comme le polynôme caractéristique de $\rho_\ell(F)$. Nous rappelons que χ_F est un polynôme unitaire de degré $2g$ et que d'après l'hypothèse de Riemann pour les courbes, si nous posons $\chi_F(x) = \prod(x - \lambda_i)$ avec $\lambda_i \in \mathbb{C}$, alors $|\lambda_i| = q^{1/2}$. De plus, il est bien connu que $\chi_F(1)$ est simplement le nombre de points $\mathbb{F}_q$-rationnels de $J(C)$.

3.0.6.1 Hypothèses de complexité

Dans tout le chapitre, nous avons $q = p^n$ avec p un nombre premier et $\mathbb{Z}_q$ (resp. $\mathbb{Q}_q$) désigne l'extension non ramifiée de degré n de $\mathbb{Z}_p$ (resp. $\mathbb{Q}_p$). Nous notons v_p la valuation p-adique sur $\mathbb{Q}_q$. Si a est un élément de $\mathbb{Z}_q$ nous notons $\bar{a}$ sa réduction modulo p dans $\mathbb{F}_q$. Nous disons que nous avons calculé un élément $x \in \mathbb{Z}_q$ à précision m si nous pouvons écrire une chaîne binaire représentant sa classe dans l'anneau quotient $\mathbb{Z}_q/p^m\mathbb{Z}_q$. Afin d'évaluer la complexité des algorithmes, nous utilisons le modèle de calcul d'un Machine à Accès Directe [104]. Nous supposons que la multiplication de deux entiers de n bits prend $O(n^\mu)$ opérations élémentaires. On a respectivement $\mu = 1 + \epsilon$ (pour n suffisamment grand), $\mu = \log_2(3)$ et $\mu = 2$ en utilisant la FFT, l'algorithme de Karatsuba et la multiplication naïve. Soient $x, y \in \mathbb{Z}_q/p^m\mathbb{Z}_q$. Par la suite, nous supposons que nous utilisons la présentation polynomiale creuse comme expliqué dans [24, pp.239]. Sous cette hypothèse, la produit xy à précision m se fait en $O(m^\mu \log(q)^\mu)$ opérations élémentaires.

3.1 Équations polynômiales tordues par le Frobenius

Dans cette section, nous introduisons une classe d'équations polynômiales à coefficients dans les p-adiques avec une action du morphisme de Frobenius sur certaines variables. Nous expliquons comment généraliser le classique algorithme de Newton afin relever dans les p-adiques des solutions de telles équations connues à petite précision. Les algorithmes décrits dans cette section serviront de brique de base pour tous les algorithmes en temps quasi-quadratiques de ce mémoire.

3.1.1 Une équation d'Artin-Schreier généralisée

Dans section, nous résumons les résultats de [77, 79], articles écrit en commun avec Reynald Lercier. Il s'agit de rappeler les principes de l'algorithme de comptage de points de Satoh et d'expliquer comment on peut en obtenir une version rapide en généralisant le classique algorithme de Newton pour calculer une approximation p-adique de la racine d'un polynôme.

Afin d'obtenir $\chi_F(1)$, Satoh propose dans [111] une nouvelle approche qui consiste à calculer l'action du morphisme de Frobenius sur les formes différentielles invariantes du relevé canonique d'une courbe elliptique ordinaire. De manière plus précise, on se donne E_0 une courbe elliptique ordinaire définie sur $\mathbb{F}_q$ de caractéristique p et de degré absolu n. Nous considérons un relevé $\tilde{E}_0$ de E_0 sur $\mathbb{Q}_q$. Une équation pour une telle courbe peut être obtenue en prenant, par exemple, un modèle de Weierstrass minimal [116, p. 172] à coefficients dans $\mathbb{Z}_q$ et se réduisant modulo p sur une équation de E_0. Le modèle de Weierstrass définit une courbe elliptique sur $\mathbb{Q}_q$ qui se réduit modulo p sur E_0. Il est alors possible de considérer une suite de courbes elliptiques isogènes $\tilde{E}_i$ avec $\tilde{E}_0$ et des morphismes $\Sigma_i : \tilde{E}_i \to \tilde{E}_{i+1}$ qui relèvent le morphisme de Frobenius $F_i : E_i \to E_{i+1}$.

Par un théorème de Lubin-Serre-Tate [85], la suite $(j(\tilde{E}_i))_{i\in\mathbb{N}}$ converge p-adiquement vers $j(E_0^{\mathrm{can}})$. En fait, dans son algorithme, Satoh [111], utilise le dual du morphisme de Frobenius qui a la propriété d'être séparable dans le cas où il agit sur une courbe ordinaire. Il est alors possible de calculer le Frobenius à la puissance n ou bien son dual comme une composition. La valeur propre du morphisme de Frobenius inversible modulo p peut être calculée comme le premier coefficient de l'action du relevé du morphisme de Frobenius à la puissance n sur le groupe formel de la courbe elliptique ou de manière alternative par l'intermédiaire de l'injection $\mathrm{End}(E_0^{\mathrm{can}}) \to \mathbb{Q}_q$ donnée par l'action d'un endomorphisme sur les formes différentielles invariantes [116, p. 163].

D'un point de vue algorithmique, la méthode de Satoh se compose de deux phases bien distinctes :

– une première phase qui consiste à calculer une approximation du j-invariant du relevé canonique de la courbe de départ jusqu'à une certaine précision p-adique ;
– une deuxième phase qui permet d'obtenir l'action du morphisme de Frobenius sur les formes différentielles invariantes du relevé canonique.

La précision p-adique à laquelle on doit faire les calculs est simplement donnée par l'hypothèse de Riemann pour les courbes qui implique que le module de la valeur propre du morphisme de Frobenius inversible modulo 2 est majoré par $\sqrt{q}$.

Il est possible d'interpréter la première étape de l'algorithme comme le calcul d'une solution p-adique d'une certaine équation polynomiale, le morphisme de Frobenius agissant sur certaines variables. En effet, le j-invariant $j(\tilde{E}_0)$ du relevé canonique de E_0 est caractérisé par les conditions

$$\Phi_p(j(\tilde{E}_0), j(\tilde{E}_0)^\sigma) = 0, \tag{3.1}$$
$$j(\tilde{E}_0) \mod p = j(E_0), \tag{3.2}$$

où $\Phi_p(X, Y)$ est le polynôme modulaire d'ordre p [117, p. 181].

Dans le cas où le morphisme de Frobenius n'agit pas sur les variables du polynôme, il existe un algorithme très efficace pour résoudre les contraintes 3.1 et (3.2) à savoir l'algorithme de Newton. Supposons que $\Phi \in \mathbb{Z}_q[X]$ soit un polynôme à coefficients dans $\mathbb{Z}_q$ et que $x_0 \in \mathbb{Z}_q$ soit une approximation p-adique d'une racine simple x de Φ à l'ordre r c'est-à-dire que l'on a $v_p(x - x_0) \geq r$ où v_p est l'habituelle valuation p-adique. On a la relation $\Phi(x_0) = 0 \mod p^r$. Écrivons un développement de Φ à l'ordre 1 en x_0 :

$$\Phi(x_0 + p^r h) = \Phi(x_0) + p^r h \Phi'(x_0) + O(p^{2r}),$$

Φ' étant le polynôme dérivé de Φ. Nous voulons trouver $h \in \mathbb{Z}_q$ tel que $\Phi(x_0) + p^r h \Phi'(x_0) = 0 \mod p^{2r}$, ce qui donne

$$h = -\frac{\Phi(x_0)}{\Phi'(x_0)} \mod p^r.$$

Nous reconnaissons dans la formule que nous venons d'écrire la version p-adique du classique algorithme de Newton. L'intérêt de l'algorithme de Newton est qu'il permet de presque doubler la précision de la solution obtenue à chaque itération. Cela en fait un algorithmes particulièrement efficace pour calculer avec les nombres p-adiques.

Cet algorithme ne s'adapte pas de manière immédiate dans le cas où l'on cherche à calculer une solution de $\Phi(X, X^\sigma) = 0$ connaissant une solution à petite précision. Dans la section 3.1.2, nous donnons une généralisation de l'algorithme de Newton permettant de résoudre ce genre de problème. Nous présentons la version multi-variable de notre algorithme.

Dans cette section, nous notons $T_{m,n}$ la complexité en temps pour calculer le produit de deux éléments de $\mathbb{Z}_q$ à précision m. De même, $S_{m,n}$ désigne le temps nécessaire pour calculer l'action du morphisme de Frobenius d'un élément de $\mathbb{Z}_q$ à précision m. Notons que d'après un résultat de [55], si $\mathbb{Z}_q$ peut se représenter avec une base Gaussienne optimale de type t alors nous avons $S_{m,n} = O(nt)$ et $T_{m,n} = O((tnm)^\mu)$. Dans ce chapitre, nous utiliserons librement la notation soft-O, afin de négliger dans les fonctions de complexités les termes logarithmiques.

3.1.2 Une généralisation de l'algorithme de Newton

Nous disons qu'une équation est une équation d'Artin-Schreier généralisée si elle peut s'écrire nous la forme

$$x^\sigma = ax + b, \text{ avec } a \in M_k(\mathbb{Z}_q), b \in \mathbb{Z}_q^k. \tag{3.3}$$

Le lemme suivant montre que l'on peut trouver rapidement une solution à une équation d'Artin-Schreier généralisée.

Lemme 2. Soit $a \in M_k(\mathbb{Z}_q), b \in \mathbb{Z}_q^k$. Alors une solution à une équation de la forme $x^\sigma = ax + b$ peut être calculée à précision m en temps $O(\log n)\max(S_{m,n}, T_{m,n})$.

Algorithm 3.1.1 `ArtinSchreierRoot`

Un algorithme pour calculer à précision m une matrice carrée A et un vecteur B, de dimension k, tels qu'une solution de l'équation $x^\sigma = ax + b \bmod p^m$ vérifie $x^{\sigma^\nu} = Ax + B \bmod p^m$.

INPUT: $a \in M_k(\mathbb{Z}_q/p^m\mathbb{Z}_q)$ et $b \in (\mathbb{Z}_q/p^m\mathbb{Z}_q)^k$, $m, \nu \in \mathbb{N}$.
OUTPUT: $A \in M_k(\mathbb{Z}_q/p^m\mathbb{Z}_q)$ et $B \in (\mathbb{Z}_q/p^m\mathbb{Z}_q)^k$.

Step 1. if $\nu = 1$ **then return** $a \bmod p^m$, $b \bmod p^m$;
Step 2. A, $B :=$ `ArtinSchreierRoot`(a, b, m, $\lfloor \frac{\nu}{2} \rfloor$) ;
Step 3. $A := AA^{\sigma^{\lfloor \frac{\nu}{2} \rfloor}} \bmod p^m$; $B := AB^{\sigma^{\lfloor \frac{\nu}{2} \rfloor}} + B \bmod p^m$;
Step 4. if $\nu \bmod 2 = 1$ **then** $A := Aa^\sigma \bmod p^m$; $B := Ab^\sigma + B \bmod p^m$;
Step 5. return A, B;

Démonstration. Nous pouvons écrire pour tout $i \in \mathbb{N}$, $x^{\sigma^i} \equiv a_i x + b_i \bmod p^w$.

Pour calculer les coefficients a_i et b_i, il suffit alors de s'inspirer de l'algorithme classique "square and multiply" en utilisant la formule de composition suivante :

$$\forall (i, i') \in \mathbb{Z}^2, \; x^{\sigma^{i+i'}} = a_i^{\sigma^{i'}} a_{i'} x + a_i^{\sigma^{i'}} b_{i'} + b_i^{\sigma^{i'}}.$$

Il est facile de voir que la complexité de cet algorithme est $\max(S_{m,n}, T_{m,n})$ ([79]). □

Soient maintenant, $x = (x_1, \dots, x_k)$ et $y = (y_1, \dots, y_k)$ deux k-uplets de variables et pour $i = 1, \dots, k$, $\phi_i(x, y) \in \mathbb{Z}_q[x_1, \dots, x_k, y_1, \dots, y_k]$ un système de k polynômes en ces variables qui définissent une fonction $\phi : \mathbb{Z}_q^k \times \mathbb{Z}_q^k \mapsto \mathbb{Z}_q^k$. Par la suite, si $\sigma \in \mathrm{Gal}(\mathbb{Q}_q/\mathbb{Q}_p)$, nous posons $y^\sigma = (y_1^\sigma, \dots, y_k^\sigma)$. Dans ce paragraphe, nous présentons un algorithme pour résoudre efficacement une équation de la forme $\phi(x, x^\sigma) = 0$. De manière plus précise, nous voudrions pouvoir trouver une solution $x \in \mathbb{Z}_q^k$ d'un système d'équation de cette forme étant donné une solution connue à petite précision.

Théorème 5. Pour $(x_0, y_0) \in \mathbb{Z}_q^k \times \mathbb{Z}_q^k$, nous notons par la suite par $\partial\phi/\partial x(x_0, y_0) \in M_k(\mathbb{Z}_q)$ (resp. $\partial\phi/\partial y(x_0, y_0) \in M_k(\mathbb{Z}_q)$) la matrice x_{ij} (resp. y_{ij}) des dérivées partielles

$$x_{ij} = (\partial\phi_i/\partial x_j)(x_0, y_0),\ (\text{resp. } y_{ij} = (\partial\phi_i/\partial y_j)(x_0, y_0)),\ 1 \leqslant i, j \leqslant k.$$

Soit $x_0 \in \mathbb{Z}_q^k$ un zéro de $\phi(x, x^\sigma) = 0 \bmod p^w$, $w \in \mathbb{N}$. Nous supposons de plus que

$$\det(\left(\frac{\partial\phi}{\partial y}\right)(x_0, x_0^\sigma)) \neq 0, \tag{3.4}$$

$$v(\left(\frac{\partial\phi}{\partial y}\right)^{-1}(x_0, x_0^\sigma)\frac{\partial\phi}{\partial x}(x_0, x_0^\sigma)) > 0 \tag{3.5}$$

et

$$v(\left(\frac{\partial\phi}{\partial y}\right)^{-1}(x_0, x_0^\sigma)\phi(x_0, x_0^\sigma)) > v(\left(\frac{\partial\phi}{\partial y}\right)(x_0, x_0^\sigma)), \tag{3.6}$$

alors une solution de l'équation $\phi(x, x^\sigma) = 0 \bmod p^m$ peut être calculée en $\tilde{O}(nm)$ opérations élémentaires.

Nous donnons l'algorithme qui a les propriétés voulues :

Algorithm 3.1.2 `NewtonLift`

Un algorithme pour calculer une solution $\phi(x, x^\sigma) \bmod p^m$, sachant une solution x_0 modulo p^{2r+1} où $r = v(\phi(x_0, x_0^\sigma)) - v((\partial\phi/\partial y)^{-1}(x_0, x_0^\sigma)v(\phi(x_0, x_0^\sigma)))$.

Input: $x_0 \in (\mathbb{Z}_q/p^{2r+1}\mathbb{Z}_q)^k$, $m \in \mathbb{N}$.
Output: x a solution de $\phi(x, x^\sigma) \bmod p^m$.

Step 1. if $m \leqslant 2r + 1$ **then return** x_0 ;
Step 2. $w := \lceil \frac{m}{2} \rceil + r$;
Step 3. $x :=$ `NewtonLift`(x_0, w);
Step 4. Relever x dans $\mathbb{Z}_q/p^m\mathbb{Z}_q$; $y := x^\sigma \bmod p^m$;
Step 5. $\Delta_x := \partial_x\phi(x, y) \bmod p^{w-r}$; $\Delta_y := \partial_y\phi(x, y) \bmod p^{w-r}$;
Step 6. $V := \phi(x, y) \bmod p^m$;
Step 7. $a, b :=$ `ArtinSchreierRoot`$(-\Delta_y^{-1}V/(p^{w-r}), -\Delta_y^{-1}\Delta_x, w - r, n)$;
Step 8. return $x + p^{w-r}(1 - a)^{-1}b$;

Démonstration. Pour la preuve de correction de l'algorithme, nous renvoyons le lecteur à [79].

L'algorithme s'appelle de manière récursive $O(\log n)$ fois avec des arguments dont la longueur est multipliée par deux à chaque appel. L'étape qui prend le plus de temps est l'appel à l'algorithme `ArtinSchreierRoot`. La complexité asymptotique de ce dernier algorithme est de $O(\log n)\max(S_{w,n}, T_{w,n})$. Pour les corps finis disposant d'une base Gaussienne Optimale, cela donne une complexité en temps de $\tilde{O}(nm)$. □

3.2 Une version rapide de l'algorithme de Mestre

Dans cette section, nous résumons les résultats de [79] article écrit avec Reynald Lercier. Dans cette partie, on considère une courbe hyperelliptique C sur $\mathbb{F}_q$ avec $q = 2^n$.

Dans le cas où le corps de base est de caractéristique 2, Jean-François Mestre a proposé dans [93, 94] une élégante généralisation de l'algorithme de Satoh permettant de calculer la cardinalité du groupe des points rationnels d'une jacobienne de courbe hyperelliptique ordinaire.

Soit C une courbe hyperelliptique ordinaire de genre g définie sur $\mathbb{F}_q$ ayant pour jacobienne $J(C)$. Il est possible de considérer l'espace de déformation local de $J(C)$, $\mathscr{A}^{loc}_{J(C)}$ qui est l'ensemble des schémas abéliens définis sur $\mathbb{Z}_q$ qui se réduisent modulo 2 sur $J(C)$. Comme dans le cas des courbes elliptiques, il existe un élément distingué $J(C)^{\mathrm{can}}_{\mathbb{Z}_q}$ de $\mathscr{A}^{loc}_{J(C)}$ appelé le relevé canonique de $J(C)$. Cet élément est caractérisé par la propriété qu'il existe un automorphisme sur $J(C)^{\mathrm{can}}_{\mathbb{Z}_q}$ qui se réduit modulo 2 sur le morphisme de Frobenius de $J(C)$. Comme $J(C)$ est par hypothèse ordinaire, d'après des considérations classiques sur l'équivalence entre la représentation rationnelle du groupe $End(J(C)^{\mathrm{can}}_{\mathbb{Z}_q})$ des endomorphismes de $J^{\mathrm{can}}_{\mathbb{Z}_q}(C)$ et la somme directe de de la représentation analytique de $End(J^{\mathrm{can}}_{\mathbb{Z}_q})$ et sa conjuguée complexe [114], il est possible de voir que l'action de dual du morphisme de Frobenius sur les formes différentielles invariantes du relevé canonique donne la moitié des valeurs propres du morphisme de Frobenius. À partir de cette description, on déduit le déroulement général de l'algorithme décrit dans cette section. Tout d'abord, calculer jusqu'à une certaine précision les valeurs de certains invariants appelés thêta constantes attachées au relevé canonique de $J(C)$ en itérant des formules de duplication. Ensuite, il s'agit de calculer l'action du relevé du dual du morphisme de Frobenius sur ces invariants et finalement de retrouver le déterminant de l'action du morphisme de Frobenius en calculant un quotient de thêta constantes.

L'algorithme peut se décrire de la manière suivante :

Phase d'initialisation. Étant donnée une courbe hyperelliptique C définie sur un corps fini $\mathbb{F}_{2^n}$, nous calculons à petite précision 2-adique les valeurs de 2^g thêta constantes $\theta^2 \left[\begin{smallmatrix} 0 \\ \varepsilon \end{smallmatrix}\right](0,\Omega)$ pour $\varepsilon \in \{0,1\}^g$, Ω étant un élément du demi-espace de Siegel correspondant à un relevé de $J(C)$ (voir le paragraphe suivant).

Phase de relèvement. En utilisant les équations de duplication de Riemann, nous avons à résoudre un système multivarié $G(x) = x^\sigma$, σ étant le morphisme de Frobenius, dans $\mathbb{Z}_q$ à précision m suffisamment grande. La solution de ce système est un vecteur de dimension $2^g - 1$. Chaque composante de cette solution est le quotient d'une thêta constante divisée par une thêta constante lié à une caractéristique fixée. Ce calcul peut être effectué à partir de la donnée des thêtas constantes à petite précision grâce à l'algorithme de relèvement donné en section 3.1.1.

Phase de norme. Il s'agit ici de calculer la norme $\mathrm{N}_{\mathbb{Z}_q/\mathbb{Z}_2}$ d'un élément $\mathbb{Z}_q$ obtenu à partir des $2^g - 1$ quotients de thêta constantes relevées. Cela donne le produit $\lambda_1 \cdots \lambda_g$ calculé à précision m des g valeurs propres de F inversibles modulo 2.

Phase LLL. Pour obtenir le polynôme caractéristique du morphisme de Frobenius χ_F de C on peut procéder selon la méthode décrite par Mestre. En premier lieu, il s'agit de construire le polynôme symétrique $P_{sym}(x)$ de degré 2^{g-1} dont les racines sont de la forme $x + q/x$ où x est le produit de g termes qui appartiennent à $\{\lambda_1, q/\lambda_1\}, \ldots, \{\lambda_g, q/\lambda_g\}$. Notons que $\eta = \lambda + 2^{gn}/\lambda$ est une de ces racines. Finalement, nous calculons les racines $P_{sym}(x)$ dans $\mathbb{C}$ et en les recombinant nous pouvons reconstruire $\chi_F(\pm x)$.

Ci-après nous détaillons chacune des phases de l'algorithme.

3.2.1 Phase d'initialisation.

Nous partons d'une courbe hyperelliptique de genre g définie sur $\mathbb{F}_{2^n}$ par l'équation affine de la forme

$$y^2 + h(x)y = q(x)h(x)$$

où $h(x)$ et $q(x)$ sont des polynômes sur $\mathbb{F}_{2^n}$ de degré $g+1$ tels que $h(x)$ ait $g+1$ racines de multiplicité 1 sur $\mathbb{F}_{2^n}$. Au prix d'une extension du corps de base et d'un changement de variable, il est toujours possible de trouver une telle paramétrisation (voir [94]).

Afin d'obtenir les thêta constantes à petite précision, nous commençons par relever dans $\mathbb{Z}_{2^n}$ le modèle affine de C $(2y + h(x))^2 = h(x)(h(x) + 2^2 q(x))$. Avec le lemme d'Hensel, il n'est pas difficile de voir que les polynômes $h(x)$ et $h(x) + 4q(x)$ sont complètement scindés sur $\mathbb{Z}_{2^n}$ et on obtient une courbe $\tilde{C}$ donnée par une équation de la forme

$$y^2 = \prod_{i=0}^{2g+1} (x - a_i) \text{ avec } a_i \in \mathbb{Z}_{2^n} \text{ et } a_{2i} \equiv a_{2i+1} \bmod 2^2.$$

Nous fixons un plongement $\phi : \mathbb{C}_2 \to \mathbb{C}$ qui permet de voir $J(\tilde{C})$ comme une variété abélienne complexe. On peut considérer une base symplectique de $H_1(C, \mathbb{Z})$ donné par les A-cycles et les B-cycles comme décrits dans [98]. Cela permet d'associer à $J(\tilde{C})$ une matrice de périodes Ω qui est un élément de $\mathbb{H}_g$ le demi-espace de Siegel de dimension g. Pour $\epsilon_1, \epsilon_2 \in \mathbb{N}^g$ et $l \in \mathbb{N}^*$, nous définissons par

$$\theta_l \left[\begin{smallmatrix}\epsilon_1\\ \epsilon_2\end{smallmatrix}\right](z, \Omega) = \sum_{n \in \mathbb{Z}^g} \exp\left[\pi i^t(n + \frac{\epsilon_1}{l})\Omega(n + \frac{\epsilon_1}{l}) + 2\pi i^t(n + \frac{\epsilon_1}{l}).(z + \frac{\epsilon_2}{l})\right]. \qquad (3.7)$$

les fonctions thêta avec caractéristique rationnelle.

Alors, les formules de Thomae-Fay (voir [98]) permettent de calculer les 2^g thêta constantes $\theta_2\left[\begin{smallmatrix}0\\e\end{smallmatrix}\right](0,\Omega)$ pour $e \in \{0,1\}^g$ à petite précision par la formule

$$\theta_2\left[\begin{smallmatrix}0\\e\end{smallmatrix}\right](0,\Omega) = \sqrt{\prod_{0\leqslant i<j\leqslant g}(a_{2i+e_i}-a_{2j+e_j})(a_{2i+1-e_i}-a_{2j+1-e_j})},$$

où $e_0 = 0$ et où $e = (e_i)_{i=1,\ldots,g}$. La racine carrée est choisie de manière arbitraire.

3.2.2 Phase de relèvement

L'objet de cette phase est de calculer le quotient de thêta constantes à précision p-adique m. Cette précision dépend de g et n.

Les formules de duplication de Riemann [36, p. 3], s'écrivent, pour $\varepsilon \in \mathbb{Z}^g$,

$$\theta_1\left[\begin{smallmatrix}0\\\varepsilon\end{smallmatrix}\right](0,2\Omega)^2 = \frac{1}{2^g}\sum_{e\in(\mathbb{Z}/2\mathbb{Z})^g}\theta_1\left[\begin{smallmatrix}0\\\varepsilon+e\end{smallmatrix}\right](0,\Omega)\theta_1\left[\begin{smallmatrix}0\\e\end{smallmatrix}\right](0,\Omega). \tag{3.8}$$

La méthode proposée par Mestre est de faire croître la précision de un à chaque étape grâce à l'utilisation des formules de duplication de Riemann. Soit $G : \mathbb{Z}_q^{2^g-1} \to \mathbb{Z}_q$ définie par $G(t_1) = 2\frac{\sqrt{t_1}}{1+t_1}$ si $g = 1$ et plus généralement si $g > 1$ par

$$G(t_1,\cdots,t_{2^g-1}) = 2\frac{\sqrt{t_1}+\sqrt{t_2}\sqrt{t_3}+\cdots+\sqrt{t_{2^g-2}}\sqrt{t_{2^g-1}}}{1+t_1+\cdots+t_{2^g-1}}, \tag{3.9}$$

qui est le déshomogénéisé de (3.8) par rapport à la variable $\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$. Posons $\tau_e^{(i)} = \theta\left[\begin{smallmatrix}0\\e\end{smallmatrix}\right](0,2^i\Omega)/\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](0,2^i\Omega)$, alors la méthode de Mestre consiste en l'itération m fois de (3.9). Cela s'écrit

$$\forall e \in \{1,\ldots,2^g-1\},\ \tau_e^{(i+1)} = G(\tau_e,\tau_{i_2}^{(i)},\tau_{i_3}^{(i)},\ldots,\tau_{i_{2^g-2}}^{(i)},\tau_{i_{2^g-1}}^{(i)}),$$

où pour tout e, les indices $i_2,\ldots,i_{2^g-1}$ sont tels que

$$\{\{0,e\},\{i_2,i_3\},\ldots,\{i_{2^g-2},i_{2^g-1}\}\} = \{\{j,j\oplus e\} \mid j \in \{1,\ldots,2^g-1\}\} \tag{3.10}$$

avec $\oplus$ qui désigne le ou exclusif de deux entiers. Il n'est pas difficile de voir que l'algorithme résultant est de complexité quasi-cubique en n si $m = O(n)$.

Nous obtenons une complexité quasi-quadratique en remarquant que

$$\forall e \in \{1,\ldots,2^g-1\},\ \tau_e^{(i+1)} = (\tau_e^{(i)})^\sigma.$$

En conservant les notations du dessus, nous calculons les racines $\tau_1,\ldots,\tau_{2^g-1}$ (telles que $\tau_e = \theta_e/\theta_0 \bmod 2^4$) du système d'équations données par,

$$\forall e \in \{1,\ldots,2^g-1\},\ \tau_e^\sigma = G(\tau_e,\tau_{i_2},\tau_{i_3},\ldots,\tau_{i_{2^g-2}},\tau_{i_{2^g-1}}). \tag{3.11}$$

Une telle équation peut être efficacement résolue grâce à l'algorithme `Newtonlift` de la section 3.1.1.

3.2.3 Phase de calcul de norme

Soit λ le produit des valeurs propres du morphisme de Frobenius inversibles modulo 2. On a

$$\lambda \equiv \mathrm{N}_{\mathbb{Z}_{2^n}/\mathbb{Z}_2}\left(\frac{2^g}{1+\tau_1+\cdots+\tau_{2^g-1}}\right) \mod 2^m.$$

Avec une base Gaussienne optimale de type t, H. Y. Kim et al. ont décrit un algorithme de type diviser pour mieux régner afin de calculer une telle norme. Cet algorithme a une complexité en temps de $\widetilde{O}(nm)$.

3.2.4 Phase de reconstruction

Elle consiste tout d'abord, en utilisant l'algorithme LLL, a obtenir le polynôme symétrique $P_{sym}(x)$ qui est le polynôme à coefficients entiers dont les racines sont de la forme $x+q/x$ où x est le produit de g termes appartenant à $\{\lambda_1, q/\lambda_1\}, \ldots, \{\lambda_g, q/\lambda_g\}$. Ensuite, nous calculons les racines complexes de $P_{sym}(x)$ et en déduisons $\chi_F(\pm x)$ ce qui est possible si ce dernier polynôme est irréductible. D'après [120] c'est toujours le cas si la jacobiennes de C est absolument simple. Une dernière vérification sur la courbe permet d'obtenir $\chi_F(x)$.

3.2.4.0.1 Réduction des réseaux. Le calcul de P_{sym} peut se faire facilement en calculant un vecteur court d'un réseau $\mathcal{L}$ donné par la matrice

$$\begin{bmatrix} \Upsilon\times M_1 & \Upsilon\times M_2 & \cdots & \Upsilon\times M_{2^{g-1}+1} & \Upsilon\times 2^m \\ 0 & 0 & \cdots & 2^{\lfloor n\times S_{2^{g-1}+1}\rfloor} & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 2^{\lfloor n\times S_2\rfloor} & \cdots & 0 & 0 \\ 2^{\lfloor n\times S_1\rfloor} & 0 & \cdots & 0 & 0 \end{bmatrix},$$

où

$$[M_i]_{i=1,\ldots,2^{g-1}+1} = \left[\ 2^{(2^{g-1}-1-i)n}\eta^i \mod 2^m \mid i\in\{0,\ldots,2^{g-1}-1\}\right]\cup[\eta^{2^{g-1}} \mod 2^m, 2^m].$$

et

$$[S_i]_{i=1,\ldots,2^{g-1}+1} = \left[\frac{(i-1)(g-2)}{2} \mid i\in\{1,\ldots,2^{g-1}\}\right]\cup\left[\frac{2^{g-1}(g-2)}{2}+1\right]$$

(les vecteurs bases sont en colonne, $\eta = \lambda + 2^{gn}/\lambda$ et Υ est une constante de grande taille). Les coefficients de P_{sym} sont les composantes d'un vecteur Π de petite norme dans $\mathcal{L}$. Des estimations asymptotiques montrent que une réduction de réseau utilisant l'algorithme LLL [71, 25] peut calculer Π si sa norme euclidienne $||\ ||_2$ (ou norme supérieure $||\ ||_1$) vérifie $||\Pi||_1 \leqslant ||\Pi||_2 \leqslant \det(\mathcal{L})^{1/\dim\mathcal{L}}$. Nous pouvons évaluer précisément, d'une part, la norme $||\ ||_1$ de Π comme fonction de n et g et d'autre part, le déterminant de $\mathcal{L}$ comme fonction de m et g et la taille de Υ (produit des éléments diagonaux). Cela donne

$$m > \frac{2^{2g}(g-2) + 2^{g+1}(g+2)}{16}\, n.$$

Nous donnons des valeurs numériques dans la table 3.1. Pour $g = 3$, par exemple, nous observons qu'une précision $m = 9n + 100$ est nécessaire.

TAB. 3.1 – Valeurs asymptotique pour la précision de m comme fonction de g et n.

g	1	2	3	4	5	6	7	8	9	10
m	$n/2$	$2n$	$9n$	$44n$	$220n$	$1088n$	$5264n$	$24896n$	$115392n$	$525824n$

3.2.5 Implémentation et résultats

Avec Reynald Lercier, nous avons implémenté notre variante de la méthode de Mestre sur un corps fini de caractéristique 2 muni d'une base Gaussienne optimale. Nous avons utilisé le système de calcul algébrique Magma version 2.10 [11]. Bien sûr, nous n'avons aucun espoir de pourvoir calculer avec cette méthode le polynôme caractéristique du morphisme de Frobenius agissant sur une courbe hyperelliptique de genre plus grand que 10 (même sur $\mathbb{F}_2$, voir la table 3.1) cependant nous donnons plus loin un exemple de genre 4. De plus, nous avons fait une implémentation soignée en langage C dans les cas spécifiques du genre 1, 2 et 3. Les résultats sont donnés dans cette section.

3.2.5.0.2 Un exemple en genre 4. Afin d'illustrer l'algorithme, nous allons calculer le polynôme caractéristique du morphisme de Frobenius d'une courbe hyperelliptique de genre 4 définie sur $\mathbb{F}_{2^4} \simeq \mathbb{F}_2[t]/(t^4 + t^3 + t^2 + t + 1)$. Le modèle affine de cette courbe est $y^2 + h(x)y + q(x)h(x) = 0$ où

$$\begin{aligned} h(x) &= (x + t^3 + t^2 + t + 1)(x + t^2)(x + t^3 + 1)(x + t + 1)(x + t^3 + t^2 + 1), \\ q(x) &= x^5 + \left(t^3 + t^2 + t + 1\right) x^4 + x^3 + t^3 x^2 + \left(t^3 + t + 1\right) x. \end{aligned}$$

Les coordonnées en x des dix points de 2 torsion d'un relevé de la courbe dans l'extension totalement non ramifiée des 2-adiques définie par le polynôme $t^4 + t^3 + t^2 + t + 1$ sont, à précision 6 données par

$$a = [-t-1,\ 4\,t^3 - 24\,t^2 - 13\,t + 15,\ -t^2,\ -28\,t^3 + 11\,t^2 - 20\,t - 28,\ -13\,t^3 + 24\,t^2 + 4\,t - 1,$$
$$-\,t^3 - 1,\ 3\,t^3 - 17\,t^2 - 8\,t - 9,\ -t^3 - t^2 - 1,\ -9\,t^3 + 27\,t^2 + 15\,t + 23,\ -t^3 - t^2 - t - 1].$$

Les formules de Thomae-Fay donnent les 15 thêta constantes $\tau_e = \theta_e/\theta_0$ qui à précision 7 sont

$$\tau = [-32\,t^3 + 16\,t^2 + 8\,t - 55, -56\,t^3 - 40\,t^2 - 32\,t + 49, -8\,t^3 + 24\,t^2 - 40\,t + 9,$$
$$-48\,t^3 + 48\,t^2 - 48\,t + 9, 48\,t^3 + 64\,t^2 - 40\,t + 1, 24\,t^3 + 24\,t^2 + 64\,t - 55, -56\,t^3 - 40\,t^2 + 56\,t - 47,$$
$$-32\,t^3 - 24\,t^2 + 56\,t - 47, 64\,t^3 + 24\,t^2 - 48\,t + 57, -40\,t^3 - 32\,t^2 - 8\,t - 15, 8\,t^3 + 64\,t^2 - 23,$$
$$48\,t^3 - 24\,t^2 + 8\,t + 41, 16\,t^3 + 24\,t^2 + 32\,t - 63, 40\,t^3 - 16\,t^2 - 40\,t - 39, -40\,t^3 - 48\,t^2 - 32\,t + 1].$$

Un appel à `NewtonLift` permet de calculer un relevé canonique de τ à précision 345. Cela donne en notation hexadécimale,

$$\begin{aligned}
\tau = [&-\mathtt{2AF4761F43EBADC244C1BC1D33E90C24141C48F828C8E05A9E7ADB00EC35D6F88BD03D0C05C445A4BFF230}\,t^3\\
&+ \mathtt{1F66D441F994F38896AF5CB90E34007AD48632BBBC09695F6E2E5A4ED676ED6752EBE9B9239EACB570F370}\,t^2\\
&+ \mathtt{1F173EE17446547549FBE4BE2CA778C1AA31398B6AB73966621BF4D4A63B45131165F0E1847B040F40E648}\,t\\
&+ \mathtt{487AD2E1552D785AC648ED52E76D6195E111BCB022D02B334512B58E067205652901ADD8E97630C49196A9},\\
&- \mathtt{CFCB3E69C51334F4EF2935E567132DB89EFBA2ED2CBF2FDF6F10269FE8C9B73C80197CA44DDDDF193B1E8}\,t^3\\
&+ \mathtt{8CEDB1F02E50ED0A7CA46A8E90C1132AC1877D6208444992E5415938DD6B33F621214AB8D5C1AE0A9DFB30}\,t^2\\
&+ \mathtt{8A9135A3DAA5E6CB9FE3D9E9C0515812C0D7700AEB85B4087C1EAAB88B81A3D686F0D9945053607014ACE0}\,t\\
&+ \mathtt{37642F247689BD3BE64A4B87406152ABACD10B896DCA2498947ED235216ACA760A9E0782D3A130E334C661}],
\end{aligned}$$

$$\lambda = \mathtt{18184F8253A78523EE4F72D801B910F4A83B8B844AAA42D2CC911C89846B4B24D5B0DB3F56FA9354B37C56D} \bmod 2^{345}.$$

Après l'étape de réduction des réseaux on obtient

$$P_{sym}(x) = x^8 + 467\,x^7 - 2^4 \cdot 25988\,x^6 - 2^8 \cdot 837798\,x^5 + 2^{12} \cdot 9084572\,x^4 +$$
$$2^{16} \cdot 417375179\,x^3 + 2^{20} \cdot 1472562912\,x^2 - 2^{24} \cdot 37930023936\,x - 2^{28} \cdot 253989847040,$$

et finalement

$$\chi_F(x) = x^8 - 2\,x^7 + 12\,x^6 - 62\,x^5 + 339\,x^4 - 2^4 \cdot 62\,x^3 + 2^8 \cdot 12\,x^2 - 2^{12} \cdot 2\,x + 2^{16}.$$

3.2.6 Conclusion

Dans cette section, nous avons décrit et prouvé la validité d'un algorithme de complxité quasi-quadratique afin de calculer le nombre de points rationnels d'une courbe hyperelliptique définie sur un corps fini de petite caractéristique. Cet algorithme semble assez efficace pour des courbes de genre plus petit que 5.

3.3 Une généralisation de l'algorithme de Mestre

Dans cette section, nous résumons les résultats de [17] écrit en commun avec Robert Carls.

Afin de généraliser l'algorithme de Mestre il est nécessaire de trouver au préalable des relations sur les thêta constantes qui décrivent le relevé canonique. Il est aisé d'obtenir de telles relations, mais il est plus difficile de décrire un ensemble "complet" de telles relations utilisables pour un algorithme de relèvement. Nous donnons un tel système complet d'équations. De plus, nous obtenons une formule de transformation, qui n'est autre qu'un autre cas particulier de relations entre des thêta constantes, qui permet de calculer des valeurs propres du morphisme de Frobenius.

Nous utilisons de manière essentielle la théorie des fonctions thêta algébriques développée par Mumford et dont les principes de base font l'objet d'un bref rappel dans la section qui suit.

3.3.1 Rappels sur la théorie de Mumford

Dans cette section, nous rappelons quelques éléments relatifs aux fonctions thêta algébriques. Pour plus de détails, nous renvoyons le lecteur à [95]. Soit A_k une variété abélienne sur un corps k. Si x est un point fermé de A_k, nous notons τ_x le morphisme de translation par x sur A_k. Soit $\mathscr{L}$ un fibré ample de degré d sur A_k. À partir de maintenant, nous supposons que d est premier à la caractéristique du corps de base ou bien que A_k est ordinaire. Il existe une isogénie $\phi_{\mathscr{L}}$ de A_k dans son dual $\hat{A}_k$ définie par $\phi_{\mathscr{L}} : A_k \to \hat{A}_k$, $x \mapsto \tau_x^* \mathscr{L} \otimes \mathscr{L}^{-1}$. Comme $\mathscr{L}$ est ample, le noyau $K(\mathscr{L})$ de $\phi_{\mathscr{L}}$ est un schéma en groupe fini. Le groupe thêta $G(\mathscr{L})$ est par définition l'ensemble de paires (x, ψ) où x est un point fermé de $K(\mathscr{L})$ et ψ est un homomorphisme de fibré en droite $\psi : \mathscr{L} \to \tau_x^* \mathscr{L}$ avec la loi de composition $(x, \psi) \circ (y, \phi) = (x + y, \tau_y^* \psi \circ \phi)$. Il est facile de voir que $G(\mathscr{L})$ est un groupe qui est une extension centrale de $K(\mathscr{L})$ par $\mathbb{G}_{m,k}$. Soit $\delta = (d_1, \ldots, d_l)$ une suite finie d'entiers tels que $d_i | d_{i+1}$, nous considérons le schéma en groupe fini $Z_d = (\mathbb{Z}/d_1\mathbb{Z})_k \times_k \ldots \times_k (\mathbb{Z}/d_l\mathbb{Z})_k$ ayant les diviseurs élémentaires donnés par δ. Le schéma en groupe fini $K_d = Z_d \times \hat{Z}_d$ où $\hat{Z}_d$ est le dual de Cartier de Z_d est isomorphe à $K(\mathscr{L})$ ([97]). Le groupe de Heisenberg de type Z_d est le schéma $\mathcal{H}_d = \mathbb{G}_{m,k} \times Z_d \times \hat{Z}_d$ muni de la loi de groupe définie sur les points fermés par $(\alpha, x_1, x_2).(\beta, y_1, y_2) = (\alpha.\beta.y_2(x_1), x_1 + y_1, x_2 + y_2)$. C'est une extension centrale de $Z_d \times \hat{Z}_d$ par $\mathbb{G}_{m,k}$. Par définition, une structure thêta Θ_d de type Z_d est un isomorphisme de $\mathcal{H}_d$ dans $G(\mathscr{L})$ qui rend le diagramme suivant commutatif :

$$\begin{tikzcd}
0 \arrow[r] & \mathbb{G}_{m,k} \arrow[r] \arrow[d, equal] & \mathcal{H}_d \arrow[r] \arrow[d, "\Theta_d"] & K_d \arrow[r] \arrow[d, "\overline{\Theta_d}"] & 0 \\
0 \arrow[r] & \mathbb{G}_{m,k} \arrow[r] & G(\mathscr{L}) \arrow[r] & K(\mathscr{L}) \arrow[r] & 0
\end{tikzcd}$$

Nous remarquons que Θ_d induit un isomorphisme, noté $\overline{\Theta}_d$ dans le précédent diagramme, de K_d dans $K(\mathscr{L})$ et en conséquence une décomposition $K(\mathscr{L}) = K_1(\mathscr{L}) \times K_2(\mathscr{L})$ où $K_2(\mathscr{L})$ est le dual de Cartier de $K_1(\mathscr{L})$. Un fait important que nous allons utiliser dans ce chapitre est qu'une structure thêta détermine à multiplication par une constante près une base des sections globales de $\mathscr{L}$ et donc détermine un plongement de A_k dans $\mathbb{P}_k^{d-1}$. Nous rappelons brièvement la construction de cette base. Tout d'abord, comme mentionné plus haut, Θ_d détermine une décomposition $K(\mathscr{L}) = K_1(\mathscr{L}) \times K_2(\mathscr{L})$. Nous rappelons que si K est un sous-groupe de $K(\mathscr{L})$, un sous-groupe de niveau $\tilde{K}$ au dessus de K est un sous-groupe de $G(\mathscr{L})$ qui est l'image d'une section de K dans $G(\mathscr{L})$. Nous définissons les sous-groupes de niveau maximaux $\tilde{K}_1$ au dessus de $K_1(\mathscr{L})$ et $\tilde{K}_2$ au dessus de $K_2(\mathscr{L})$ comme l'image par Θ_d des sous-groupes $(1, x, 0)_{x \in Z_d}$ et $(1, 0, y)_{y \in \hat{Z}_d}$ de $\mathcal{H}_d$. Soit B_k le quotient de A_k par $K_2(\mathscr{L})$ et $\pi : A_k \to B_k$ la projection naturelle. La théorie de descente de Grothendieck nous dit que la donnée de $\tilde{K}_2$ est équivalente à la donnée d'un couple $(\mathscr{L}_0, \lambda)$ où $\mathscr{L}_0$ est un fibré en droite ample de degré 1 sur B_k et λ est un isomorphisme $\lambda : \pi^*(\mathscr{L}_0) \to \mathscr{L}$. Soit s_0 l'unique section globale de $\mathscr{L}_0$ à une constante multiplicative près et soit $s = \lambda(\pi^*(s_0))$.

Proposition 1. Pour tout $i \in Z_d$, soit $(x_i, \psi_i) = \Theta_d((1, i, 0))$. Nous posons $\vartheta_i^{\Theta_d} = (\tau^*_{-x_i} \psi_i(s))$. Les éléments $(\vartheta_i^{\Theta_d})_{i \in Z_d}$ forment une base des sections globales de $\mathscr{L}$ qui est uniquement déterminée à une constante multiplicative près indépendante de i par la donnée de Θ_d.

Cette dernière proposition n'est qu'une reformulation de [95, Th. 2], qui dit que l'espace des sections globales de $\mathscr{L}$ est l'unique représentation irréductible de poids 1 sous l'action du groupe thêta.

Du fait de la précédente proposition, il est possible d'identifier le k-espace vectoriel des section globales de $\mathscr{L}$ avec l'espace $k[Z_d]$ des fonctions de Z_d dans k. Il y a une action de $\mathcal{H}_d$ sur $k[Z_d]$ donnée, pour tout $(\alpha, y_1, y_2) \in \mathcal{H}_d$, $f \in k[Z_d]$ et $x \in Z_d$, par

$$(\alpha, y_1, y_2).f(x) = \alpha.y_2(x) f(x + y_1). \tag{3.12}$$

Cette action est compatible via Θ_d avec l'action naturelle de $G(\mathscr{L})$ sur les section globales de $\mathscr{L}$.

Soit $(\vartheta_i^{\Theta_d})_{i \in Z_d}$ une base des sections globales de $\mathscr{L}$ déterminée par une structure thêta Θ_d et soit x un point fermé de A_k. Notons par $\mathscr{O}_{A_k}$ le faisceau structural de A_k et soit $\rho : \mathscr{O}_{A_k,x} \to k'$ le morphisme d'évaluation à valeur dans le corps résiduel de x. Nous pouvons choisir un isomorphisme $\xi : \mathscr{L}_x \simeq \mathscr{O}_{A_k,x}$. Pour tout $i \in Z_d$, l'évaluation d'une section de $\vartheta_i^{\Theta_d}$ en x est par définition $\vartheta_i^{\Theta_d}(x) = \rho \circ \xi(\vartheta_i^{\Theta_d})$. Le point projectif qui en résulte $(\vartheta_i^{\Theta_d}(x))_{i \in Z_d}$ défini sur $\overline{k}$ ne dépend pas du choix de l'isomorphisme ξ.

En appliquant cette dernière construction au point fermé 0 de A_k, nous pouvons associer à tout triplet $(A_k, \mathscr{L}, \Theta_d)$ un point de $\mathbb{P}^d_k$, donné par ses coordonnées projectives $(\vartheta_i^{\Theta_d}(0))_{i \in Z_d}$. Ce point est appelé le point des thêta constantes associé à $(A_k, \mathscr{L}, \Theta_d)$.

3.3.2 Relations thêta tordues par le Frobenius

Dans toute cette partie, nous utilisons les notations suivantes. Soit $\mathbb{F}_q$ le corps fini de caractéristique $p > 0$ ayant q éléments. Soit $\mathbb{Z}_q$ l'anneau des vecteurs de Witt à coefficients dans $\mathbb{F}_q$ et soit $\sigma \in \mathrm{Aut}(\mathbb{Z}_q)$ le relèvement canonique du Frobenius absolu agissant sur $\mathbb{F}_q$. Soit A un schéma abélien de dimension relative g sur $\mathbb{Z}_q$. Nous désignons $Z_n = (\mathbb{Z}/n\mathbb{Z})^g$ pour $n \geq 1$. Nous notons par $\mathscr{L}$ un fibré symétrique ample de degré 1 sur A. Nous supposons que A a bonne réduction et que A est le relevé canonique de sa réduction $A_{\mathbb{F}_q}$.

Nous considérons tout d'abord le cas $p > 2$. Supposons que soit donnée une structure thêta Θ_2 de type Z_2 pour $\mathscr{L}^2$ ainsi qu'un isomorphisme

$$Z_{p,\mathbb{Z}_q} \xrightarrow{\sim} A[p]^{\mathrm{et}} \tag{3.13}$$

où $A[p]^{\mathrm{et}}$ désigne le quotient étale maximal de $A[p]$. D'après [18, Th.2.2], il existe une thêta structure symétrique canonique Θ_p de type Z_p pour le fibré en droite $\mathscr{L}^p$ qui est uniquement déterminée par l'isomorphisme (3.13). Nous définissons $\Theta_{2p} = \Theta_2 \times \Theta_p$ comme la thêta structure semi-canonique produit de type Z_{2p} pour $\mathscr{L}^{2p}$.

Nous supposons que $p = 2$. On se donne un isomorphisme

$$Z_{4,\mathbb{Z}_q} \xrightarrow{\sim} A[4]^{\mathrm{et}} \tag{3.14}$$

où $A[4]^{\mathrm{et}}$ désigne le quotient étale maximal de $A[4]$. Dans ce cas, il existe une structure thêta canonique symétrique Θ_4 de type Z_4 qui est déterminée de manière unique par l'isomorphisme (3.14).

Le théorème suivant peut être montré pour tout premier $p > 0$. Soient $(a_u)_{u \in Z_{2p}}$ les thêta constantes pour la thêta structure Θ_{2p}. Soit S l'ensemble de tous les 4-uplets $(x, y, v, t) \in Z_{2p}^4$ tels que les ensembles $\{x + y, x - y\}$ et $\{v + pt, v - pt\}$ soient égaux et contenus dans Z_p. Ici, nous considérons Z_p comme un sous-groupe de Z_{2p} via l'application $j \mapsto 2j$.

Théorème 6. Il existe un $\omega \in \mathbb{Z}_q^*$ tel que pour tout $(x, y, v, t) \in S$ on ait

$$\sum_{z \in Z_2} a_{x+z} a_{y+z} = \omega \sum_{u \in Z_{2p}} a_{v+pu} a_{t+u}^{\sigma^2}.$$

Nous considérons que Z_2 est un sous-groupe de Z_{2p} via l'application $j \mapsto pj$. Pour une preuve du théorème, voir [17].

Exemple $g = 1$, $p = 3$:

$$\begin{aligned}
a_1 a_0 + a_2 a_3 &= \omega(a_1 a_0^{\sigma^2} + a_2 a_3^{\sigma^2} + 2a_1 a_2^{\sigma^2} + 2a_2 a_1^{\sigma^2}) \\
a_2 a_0 + a_1 a_3 &= \omega(2a_1 a_1^{\sigma^2} + a_2 a_0^{\sigma^2} + a_1 a_3^{\sigma^2} + 2a_2 a_2^{\sigma^2}) \\
a_3 a_3 + a_0 a_0 &= \omega(2a_0 a_2^{\sigma^2} + a_3 a_3^{\sigma^2} + 2a_3 a_1^{\sigma^2} + a_0 a_0^{\sigma^2}) \\
a_0 a_3 + a_3 a_0 &= \omega(a_0 a_3^{\sigma^2} + 2a_3 a_2^{\sigma^2} + a_3 a_0^{\sigma^2} + 2a_0 a_1^{\sigma^2}).
\end{aligned}$$

3.3.3 Des équations de Riemann pour le niveau $2p$

Nous gardons les notations de la sections précédente. Nous supposons que $p > 2$. Soit Θ_{2p} une thêta structure symétrique de type Z_{2p} pour le fibré en droite $\mathscr{L}^{2p}$. Nous notons $(a_u)_{u \in Z_{2p}}$ les thêta constantes pour la thêta structure Θ_{2p}.

L'analogue en dimension supérieure des classiques équations de Riemann de niveau $2p$ est donné par le théorème suivant (comparer avec [95, §3]). Nous considérons les quadruplés $(v_1, w_1, x_1, y_1), (v_2, w_2, x_2, y_2) \in Z_{2p}^4$ comme équivalents s'il existe une matrice de permutation $P \in \mathrm{Mat}_4(\mathbb{Z})$ telle que

$$(v_1 + w_1, v_1 - w_1, x_1 + y_1, x_1 - y_1) = (v_2 + w_2, v_2 - w_2, x_2 + y_2, x_2 - y_2)P.$$

Théorème 7. Pour des quadruplés équivalents $(v_1, w_1, x_1, y_1), (v_2, w_2, x_2, y_2) \in Z_{2p}^4$ et pour tout $s \in Z_2$, nous avons l'égalité suivante

$$\sum_{t \in Z_2} a_{v_1+t} a_{w_1+t} a_{x_1+t+s} a_{y_1+t+s} = \sum_{t \in Z_2} a_{v_2+t} a_{w_2+t} a_{x_2+t+s} a_{y_2+t+s}.$$

Dans cet énoncé, Z_2 est considéré comme un sous-groupe de Z_{2p} via l'application $j \mapsto pj$. Nous remarquons qu'en prenant $s = 0$ dans l'équation ci-dessus, on retrouve l'équation (C) de [95, §3]. L'exemple suivant montre que l'on obtient un nombre "suffisant" d'équations bien que 4 ne divise par le niveau $2p$ pour $p > 2$.

Exemple $g = 1$, $p = 3$:

$$\begin{aligned} 0 = &\quad a_1 a_0^2 a_3 - 2a_1^2 a_2^2 + a_2 a_0 a_3^2 \\ 0 = &\quad a_2 a_0^3 - a_2^4 - a_1^4 + a_1 a_3^3 \end{aligned}$$

3.3.4 Calcul des $2p$-thêta constantes

Nous conservons les notations des sections qui précèdent. Dans la suite, nous supposons que la variété abélienne A est a réduction ordinaire et que A est le relevé canonique de sa réduction $A_{\mathbb{F}_q}$. Soit $p > 2$. Nous supposons donnée une thêta structure produit canonique $\Theta_{2p} = \Theta_2 \times \Theta_p$. Comme d'habitude, nous notons $(a_u)_{u \in Z_{2p}}$ les thêta constantes pour la thêta structure Θ_{2p}. Nous pouvons supposer qu'il existe $v \in Z_2$ tel que a_v soit inversible dans $\mathbb{Z}_q$. Ici Z_2 est considéré comme un sous-groupe de Z_{2p} via l'application $j \mapsto pj$. Soit I l'idéal de l'anneau de polynômes multivariés $\mathbb{F}_q[x_u | u \in Z_{2p}]$ engendré par les relations du théorème 7 prises modulo p, ainsi que par les relations de symétrie $x_u = x_{-u}$ pour tout $u \in Z_{2p}$. Soit J l'image de I par l'application de spécialisation

$$\mathbb{F}_q[x_u | u \in Z_{2p}] \to \mathbb{F}_q[x_u | u \in Z_{2p}, 2u \neq 0], \quad x_u \mapsto \begin{cases} \frac{a_u}{a_v}, & u \in Z_2 \\ \frac{x_u}{a_v}, & \text{sinon} \end{cases}.$$

Théorème 8. L'idéal J définit une variété algébrique de dimension 0.

Pour une preuve de ce théorème, voir [17].
Soit $f(x) \in \mathbb{F}_q[x]$ tel que

$$\mathbb{F}_q[x_u | u \in Z_{2p}, 2u \neq 0]/J \cong \mathbb{F}_q[x]/(f).$$

Les thêta constantes $(a_u)_{u \in Z_{2p}}$ donnent un élément $z \in \overline{\mathbb{F}}_q$ tel que $f(z) = 0$. Il est possible de montrer que z est un zéro de f de multiplicité p^{2g}. Le théorème 8 permet de retrouver les thêta constantes $(a_u)_{u \in Z_{2p}}$ sur une extension de $\mathbb{F}_q$ à partir de la connaissance de sa 2-torsion. En utilisant les formules de Thomae et un algorithme de base de Groebner particulier (voir la section 4.3.1), il est possible de calculer les thêta constantes de niveau $2p$ correspondant à une courbe hyperelliptique ordinaire définie sur $\mathbb{F}_q$.

3.3.5 Une formule des traces généralisée

Soit A un schéma abélien sur $\mathbb{Z}_q$. Nous supposons que A est à réduction ordinaire et que A est le relevé canonique de sa réduction $A_{\mathbb{F}_q}$. Supposons que $\Theta_{2^\nu p} = \Theta_{2^\nu} \times \Theta_p$ soit une thêta structure produit semi-canonique sur $\mathbb{Z}_q$ de type $Z_{2^\nu p}$ pour $\mathscr{L}^{2^\nu p}$. Soient $(a_u)_{u \in Z_{2^\nu p}}$ les thêta constantes pour la thêta structure $\Theta_{2^\nu p}$.

Soit $F \in \mathrm{End}_{\mathbb{F}_q}(A_{\mathbb{F}_q})$ le morphisme de Frobenius absolu de $A_{\mathbb{F}_q}$ et soit ℓ un premier différent de la caractéristique p du corps de base $\mathbb{F}_q$. Soit $T_\ell(A_{\mathbb{F}_q})$ le module de Tate ℓ-adique de $A_{\mathbb{F}_q}$. Rappelons que le module de Tate ℓ-adique est un $\mathbb{Z}_\ell$-module libre de rang $2g$. Le morphisme de Frobenius absolu F induit une application $\mathbb{Z}_\ell$-linéaire $\rho_\ell(F)$ sur $T_\ell(A_{\mathbb{F}_q})$ qui correspond, une fois une base de $T_\ell(A_{\mathbb{F}_q})$ choisie, à une matrice M_F de dimension $(2g \times 2g)$ à coefficients dans $\mathbb{Z}_\ell$. Du fait de la réduction ordinaire, nous savons que M_F a exactement g valeurs propres $\pi_1, \dots, \pi_g$, qui sont inversibles modulo p [28, Ch.V].

Théorème 9. Supposons que Θ_{2^ν} est défini sur $\mathbb{Z}_q$. Alors le produit $\pi_1 \cdot \ldots \cdot \pi_g$ est un élément de $\mathbb{Z}_q$ et nous avons

$$\pi_1 \cdot \ldots \cdot \pi_g = \mathrm{N}_{\mathbb{Q}_q/\mathbb{Q}_p} \left(\frac{\sum_{u \in Z_{2^\nu}} a_u}{\sum_{u \in Z_{2^\nu p}} a_u} \right). \tag{3.15}$$

Ici, Z_{2^ν} est considéré comme un sous-groupe de $Z_{2^\nu p}$ via l'application $j \mapsto pj$.

Pour une preuve de ce résultat, nous renvoyons le lecteur vers [17].

3.3.6 Des résultats de complexité

Nous expliquons dans cette section comment on peut utiliser les formules données dans les sections 3.3.2, 3.3.3, 3.3.5 afin de calculer le nombre de points rationnels sur une jacobienne de courbe hyperelliptique ordinaire définie sur un corps fini de caractéristique impaire. Supposons que nous ayons choisi un premier $p > 2$ et un entier $g \geq 1$.

Théorème 10. Soit C une courbe hyperelliptique de genre g sur le corps fini $\mathbb{F}_q$ de caractéristique p telle que la jacobiennes $J(C)$ soit ordinaire et absolument simple. Soit ν un entier plus grand que 3. Nous supposons que la 2^ν-torsion de $J(C)$ soit définie sur $\mathbb{F}_q$, alors il existe un algorithme pour calculer le nombre de points $\mathbb{F}_q$-rationnels $\#C(\mathbb{F}_q)$ de la courbe C ayant pour complexité asymptotique $O(n^{2+o(1)})$ en temps et $O(n^2)$ en espace où $n = \log(\#\mathbb{F}_q)$.

À partir de ce théorème il est possible de déduire le

Corollaire 3. Soit C une courbe hyperelliptique de genre g sur le corps fini $\mathbb{F}_q$ de caractéristique p telle que la jacobienne $J(C)$ soit ordinaire et absolument simple. Il existe un algorithme pour calculer le nombre de points $\mathbb{F}_q$-rationnels $\#C(\mathbb{F}_q)$ de la courbe C ayant pour complexité asymptotique $O(n^{2+o(1)})$ en temps et $O(n^2)$ en espace où $n = \log(\#\mathbb{F}_q)$.

Pour une preuve de ces résultats, voir [17].

3.3.7 Exemples

Dans [17], nous donnons deux versions de l'algorithme : une version dont on prouve qu'elle remplit les conditions du théorème 10 et une version heuristique qui a un très bon comportement en pratique. La version prouvée de l'algorithme nécessite la résolution d'un système algébrique qui la rend délicate à implémenter de manière efficace. Nous avons implémenté la version heuristique de notre algorithme dans le cas du genre 1 et du genre 2 (voir [20]). Pour l'implémentation du genre 2, en utilisant un algorithme de résolution de bases de Groebner spécifique à notre problème, il est possible de calculer efficacement une solution du système algébrique de la phase d'initialisation.

3.3.7.0.3 Un exemple en genre 1 **et caractéristique** 5. Soit $\mathbb{F}_{5^8}$ représenté comme le quotient $\mathbb{F}_5[X]/(P)$ avec $P(X) = X^8 + X^4 + 3X^2 + 4X + 2$ et soit u l'image de X dans $\mathbb{F}_{5^8}$ via l'isomorphisme précédent. Soit E la courbe elliptique ordinaire donnée par l'équation de Weierstrass

$$y^2 = x^3 + x^2 + 3x.$$

Après l'étape d'initialisation, nous obtenons les six thêta constantes suivantes

$$[1, 4, u^{32552}, u^{309244}, u^{211588}, u^{32552}].$$

Nous considérons $\mathbb{Z}_{5^8}$ comme l'extension non ramifiée des entiers 5-adiques $\mathbb{Z}_5$ définie par le polynôme $X^8 + X^4 + 3X^2 + 4X + 2$ et notons par z l'image de X dans $\mathbb{Z}_{5^8}$.

Après la phase de relèvement, nous obtenons les thêta constantes du relevé canonique à précision 5

$$\begin{aligned}
&[1, -1460z^7 - 10z^6 - 785z^5 + 715z^4 - 555z^3 + 420z^2 - 1035z - 1116,\\
&-1449z^7 - 819z^6 + 396z^5 + 746z^4 + 1108z^3 + 648z^2 + 546z - 1189,\\
&1438z^7 - 1497z^6 + 1548z^5 - 777z^4 + 354z^3 - 876z^2 + 998z + 1029,\\
&1449z^7 + 819z^6 - 396z^5 - 746z^4 - 1108z^3 - 648z^2 - 546z - 868,\\
&-1504z^7 + 101z^6 + 741z^5 + 591z^4 - 957z^3 - 492z^2 - 1109z - 834]
\end{aligned}$$

où z est un générateur.

Après la phase de norme, nous obtenons que le nombre de points rationnels de E est 1054.

3.3.7.0.4 Un exemple en genre 2 **et caractéristique** 3 Soit $\mathbb{F}_{3^{28}}$ représenté par le quotient $\mathbb{F}_3[X]/(P)$ avec

$$P(X) = X^{28} + 2X^{14} + X^{13} + X^{12} + 2X^{11} + X^{10} + X^9 + X^8 + 2X^6 + 2X^4 + X^3 + 2$$

et soit w l'image de X dans $\mathbb{F}_{5^8}$ via cet isomorphisme.

Soit H la courbe hyperelliptique ordinaire donnée par l'équation affine

$$\begin{aligned}
y^2 \quad = \quad & x^6 + (w^{18} + w^{17} + w^{16} + w^{11} + w^{10} + w^9 + w^8 + w^7 + 2w^5 + 2w^2 + w)x^5\\
&+(w^{19} + 2w^{17} + 2w^{16} + w^{13} + w^{11} + w^{10} + 2w^8 + 2w^7 + w^6 + 2w^4 + w + 2)x^4\\
&+(2w^{19} + 2w^{18} + 2w^{17} + 2w^{15} + 2w^{14} + 2w^{12} + 2w^{11} + 2w^{10} + 2w^9 + w^7 + 2w^6\\
&\quad +w^5 + 2w^4 + w^3 + w + 1)x^3\\
&+(w^{19} + 2w^{18} + 2w^{16} + 2w^{13} + w^{12} + w^{10} + 2w^9 + w^8 + w^6 + 2w^2 + 1)x^2\\
&+(w^{19} + 2w^{18} + w^{17} + 2w^{15} + 2w^{14} + w^{13} + w^{12} + w^{11} + 2w^9 + w^8 + w^6\\
&\quad +2w^5 + 2w^4 + w^3 + 2w^2 + 2)x\\
&+w^{19} + 2w^{16} + w^{15} + w^{14} + w^{12} + 2w^8 + w^7 + w^6 + w^4 + 2w^3 + w^2 + w + 1
\end{aligned}$$

Tout d'abord, nous calculons les thêta constantes de niveau 2

$$
\begin{aligned}
x_{00} &= w^{19} + w^{18} + 2 * w^{15} + w^{14} + w^{12} + 2w^{10} + w^7 + 2w^6 + 2w^5 + 2w^4 + w^3 + w + 2 \\
x_{03} &= w^{19} + 2w^{17} + 2w^{16} + 2w^{15} + 2w^{14} + 2w^{13} + w^{12} + 2w^{11} + 2w^{10} + 2w^9 + 2w^8 + w^7 + \\
&\quad w^6 + 2w^5 + w^4 + w^3 + 2w^2 + 2w + 2 \\
x_{30} &= w^{19} + 2w^{18} + w^{17} + w^{16} + 2w^{15} + 2w^{14} + 2w^{13} + w^{12} + 2w^{11} + 2w^{10} + 2w^9 \\
&\quad +2w^7 + 2w^3 + w^2 + 2 \\
x_{33} &= 2w^{19} + 2w^{18} + w^{17} + 2w^{15} + 2w^{13} + 2w^{12} + w^{10} + 2w^9 + w^8 + w^6 + 2w^4 + 2w^3 + w^2 + 2w + 1.
\end{aligned}
$$

Après l'étape de base de Groebner, nous obtenons la liste suivante de thêta constantes

$$
\begin{aligned}
0 &= x_{01} + w^{18} + w^{16} + w^{15} + 2w^9 + w^8 + w^7 + w^6 + 2w^5 + w^4 + 2 \\
0 &= x_{02} + w^{19} + 2w^{17} + 2w^{16} + 2w^{15} + w^{14} + w^{13} + w^{12} + w^{11} + w^{10} + w^9 + \\
&\quad w^7 + 2w^5 + w^4 + w^3 + w^2 + w + 2 \\
0 &= x_{10} + 2w^{19} + 2w^{18} + w^{17} + 2w^{14} + 2w^{13} + 2w^{12} + w^{11} + w^{10} + w^9 + 2w^8 \\
&\quad +2w^6 + 2w^4 + w^3 + 2w^2 + 2 \\
0 &= x_{11} + 2w^{19} + w^{16} + w^{15} + 2w^{14} + 2w^{12} + 2w^{11} + 2w^{10} + 2w^9 + w^7 + w^5 \\
&\quad +w^4 + w^3 + 2w^2 + 2w + 2 \\
0 &= x_{12} + w^{19} + 2w^{18} + 2w^{17} + w^{16} + 2w^{15} + w^{14} + w^{13} + w^{12} + 2w^{10} + 2w^9 + w^8 + \\
&\quad 2w^7 + 2w^6 + w^4 + 2w^3 + 2w^2 + 2w + 1 \\
0 &= x_{13} + w^{18} + w^{17} + 2w^{14} + 2w^{13} + w^9 + 2w^6 + 2w^5 + 1 \\
0 &= x_{20} + w^{19} + w^{18} + 2w^{16} + w^{15} + w^{14} + w^{13} + w^{12} + w^{11} + 2w^{10} + w^9 + 2w^7 \\
&\quad +2w^6 + w^4 + w^3 + w + 2 \\
0 &= x_{21} + w^{19} + w^{17} + w^{16} + w^{15} + 2w^{14} + 2w^{12} + w^{10} + w^5 + w^3 + w^2 + w + 2 \\
0 &= x_{22} + 2w^{19} + w^{17} + 2w^{16} + 2w^{15} + w^{13} + w^{12} + 2w^{11} + 2w^{10} \\
&\quad +2w^9 + w^8 + 2w^7 + 2w^5 + w^4 + w^2 + w + 1 \\
0 &= x_{23} + w^{18} + 2w^{14} + w^{12} + 2w^{11} + 2w^{10} + w^8 + w^6 + w^5 + w^2 + w + 1 \\
0 &= x_{31} + w^{18} + w^{17} + w^{16} + 2w^{15} + 2w^{13} + 2w^{11} + w^9 + w^8 + w^7 + 2w^4 + 2w^3 + 2w^2 + 2 \\
0 &= x_{32} + 2w^{19} + 2w^{18} + 2w^{17} + 2w^{16} + w^{15} + 2w^{14} + w^{13} + w^{12} + w^{11} + w^9 + w^7 + w^6 + 2w^2 + w \\
0 &= x_{41} + w^{18} + 2w^{16} + 2w^{15} + 2w^{13} + w^{12} + w^{11} + w^{10} + 2w^9 + w^8 + w^7 + 2w^6 + w^4 + 2w \\
0 &= x_{42} + 2w^{16} + 2w^{14} + 2w^{12} + 2w^{10} + w^9 + 2w^8 + 2w^6 + 2w^5 + 2w^4 + w^3 + w^2 + w + 2 \\
0 &= x_{51} + 2w^{18} + w^{17} + 2w^{16} + 2w^{15} + 2w^{13} + w^{11} + w^{10} + w^9 + 2w^8 + w^7 + 2w^6 + \\
&\quad 2w^5 + 2w^4 + 2w^3 + w^2 + 2w \\
0 &= x_{52} + 2w^{17} + 2w^{16} + 2w^{15} + w^{14} + 2w^{12} + w^{10} + 2w^9 + 2w^7 + w^6 + w^5 + 2w^4 + w^3 + 2w^2 + w
\end{aligned}
$$

Après la phase de norme, nous obtenons à précision 56 le produit des valeurs propres du morphisme de Frobenius qui sont des unités modulo 3

$$2023954210169141309384885 32.$$

À partir de là, il est facile de retrouver le polynôme χ_F qui est

$$\chi_F(X) = X^4 + 19612X^3 - 4108934426X^2 + 68382815672412X + 121576654590569288 01.$$

3.3.8 Conclusion

Nous avons décrit un algorithme ayant une complexité quasi-quadratique en temps et quadratique en espace par rapport à la taille du corps de base pour calculer le nombre de points rationnels d'une courbe hyperelliptique dont la jacobienne est ordinaire et absolument simple.

Nous avons donné deux versions de notre algorithme, une première avec une borne de complexité prouvée mais difficile à implémenter de manière efficace et une deuxième heuristique ayant un très bon comportement en pratique.

3.4 Travaux en cours

Dans cette section, nous présentons des travaux correspondant à des articles non encore acceptés pour publications.

3.4.1 Un algorithme utilisant la cohomologie à support compact

Dans cette section, nous résumons les résultats de [22], article écrit avec Gweltaz Chatel non encore publié. L'objet de cet article est de décrire et d'évaluer la complexité d'un algorithme de comptage de points utilisant la cohomologie de Monsky-Washnitzer à support compact.

Dans toute cette section, k désigne un corps fini de caractéristique impaire, $W(k)$ est l'anneau des vecteurs de Witt à coefficients dans k et K est le corps des fractions de $W(k)$.

3.4.1.1 Une description géométrique

Par la suite, nous regardons le cas des courbes hyperelliptiques qui constitue le cas le plus simple d'une famille de courbes ayant un genre arbitraire. Soit k un corps fini de caractéristique impaire. Soit C_k le modèle affine d'une courbe hyperelliptique de genre g sur k donné par une équation de la forme $Y^2 = \prod_{i=1}^{2g+1}(X - \overline{\lambda}_i)$ où $\overline{\lambda}_i \in k$. Soit $\pi_k : C_k \to \mathbb{A}^1_k$ la projection le long de l'axe des Y. Si nous notons par U_k le lieu étale de π_k et par V_k son image, nous avons un diagramme

$$\begin{tikzcd} C_k & U_k \arrow[l, hook'] \arrow[d, "\pi_k"] \\ \mathbb{A}^1_k \arrow[from=u, "\pi_k"] & V_k \arrow[l, hook'] \end{tikzcd}, \tag{3.16}$$

où les applications horizontales sont des immersions ouvertes et où π_k est fini et étale au dessus de V_k. Soient A_k et B_k les anneaux de coordonnées associés à U_k et V_k. Par [34, Th.6], il est possible de relever le diagramme (3.16) en un diagramme

$$\begin{tikzcd} C & U \arrow[l, hook'] \arrow[d, "\pi"] \\ \mathbb{A}^1_{W(k)} \arrow[from=u, "\pi"] & V \arrow[l, hook'] \end{tikzcd}, \tag{3.17}$$

de $W(k)$-schémas lisses où les applications horizontales sont des immersions ouvertes et π est finie et étale sur V. Soient A et B les anneaux de coordonnées de U et V respectivement. Soit $\Lambda_k = \{\overline{\lambda}_1, \ldots, \overline{\lambda}_{2g+1}, \overline{\infty}\}$ le complément de $V_k(\overline{k})$ dans $\mathbb{P}^1_k(\overline{k})$. Nous pouvons supposer que $B = W(k)[t, (t-\lambda_1)^{-1}, \ldots, (t-\lambda_{2g+1})^{-1}]$ où t est une indéterminée et où $\lambda_i \in W(k)$ relève $\overline{\lambda}_i$ pour $i = 1, \ldots, 2g+1$. Soit $\Lambda = \{\lambda_1, \ldots, \lambda_{2g+1}, \infty\}$, $A_K = A \otimes_{W(k)} K$ et $B_K = B \otimes_{W(k)} K$. Par le théorème de descente finie étale (c.f. [122, Cor2.6.6]) nous avons $H^1_{MW,c}(U_k/K) = H^1_{MW,c}(V, \pi_* A^\dagger_K)$. Par la suite, nous considérons toujours $A^\dagger_K$ avec sa structure de $B^\dagger_K$-module donnée par π.

Il existe une connexion de Gauss-Manin $A_K^\dagger$ et par définition le calcul d'une base de la cohomologie de la courbe C_k se ramène au calcul des sections horizontales pour cette connexion.

3.4.1.2 L'espace M_c

Dans cette section, nous décrivons un module à connexion qui va nous intéresser par le suite.

Le calcul de l'espace $H^1_{MW,c}(U_k/K)$ se ramène au calcul du module de de Rham des formes analytiques à support compact. D'après le théorème de descente finie étale [122, Cor.2.6.6], cet espace de fonctions analytiques est donné par

$$M_c = A_K^\dagger \otimes_{B_K^\dagger} B_c.$$

Pour $\lambda \in \Lambda_0$, soit $M_{c,\lambda} = A_K^\dagger \otimes_{B_K^\dagger} \tilde{R}_{\lambda,c}$ et soit $M_{c,\infty} = A_K^\dagger \otimes_{B_K^\dagger} R_{\infty,c}$. Nous avons

$$M_c = \bigoplus_{\lambda\in\Lambda} M_{c,\lambda}.$$

Un élément de $M_{c,\lambda}$ peut donc s'écrire comme

$$m_\lambda = \sum_{j=0,1} Y^j \sum_{\ell=0}^{\infty} b_{j,\ell}^\lambda (t-\lambda)^\ell.$$

avec $b_{j,\ell}^\lambda \in K$ et avec $b_{j,0}^\infty = 0$. Nous conservons la convention de notation $(t-\infty) = t^{-1}$. Le $B_K^\dagger$-module fini M_c est muni d'une connexion donnée par

$$\nabla_c : M_c \to M_c \otimes_{B_K^\dagger} \Omega^1_{B_K^\dagger},$$

$$m \otimes g_c \mapsto \nabla_{GM}(m) \otimes g_c + m \otimes \frac{\partial}{\partial t} g_c.dt,$$

où ∇_{GM} est la connexion du Gauss-Manin naturelle sur $A_K^\dagger$. Dans notre cas, cette connexion naturelle est donnée par la dérivation partielle par rapport à Y agissant sur le $B_K^\dagger$-module $A_K^\dagger$. Par définition, l'espace $H^1_{MW,c}(V, \pi_* A_K^\dagger)$ est le noyau de ∇_c. En conséquence de quoi, nous avons à calculer une base des solutions de l'équation différentielle

$$\nabla_c(m_c) = 0 \tag{3.18}$$

définie sur M_c. Il convient de remarquer que par des résultats classiques (voir for exemple [70]) la dimension de cet espace est égal à $2g$ plus le nombre de points que l'on a enlevé de la droite affine. Dans notre cas, cela donne $4g+1$.

La méthode dite locale de [22] repose sur la remarque que l'équation (3.18) localement en λ peut s'interpréter comme une équation différentielle linéaire in-homogène que l'on peut résoudre si l'on connaît suffisamment de termes d'une solution globale. C'est le contenu de la

Proposition 2. Soit $m_c = (m_{\lambda_1}, \ldots, m_{\lambda_{2g+1}}, m_\infty) \in H^1_{MW,c}(V, \pi_* A_K^\dagger)$. Pour $\lambda \in \Lambda$, soit $\nabla_{GM,\lambda}$ l'action de la connexion de Gauss-Manin sur la composante locale en λ d'un élément de M_c. Pour tout $\lambda \in \Lambda$, il existe un unique $u = u_0 + Yu_1$, avec $(u_0, u_1) \in (K((t-\lambda)))^2$ tel que m_λ soit une solution de l'équation différentielle non homogène :

$$\frac{\partial}{\partial t} m_\lambda + \nabla_{GM,\lambda} m_\lambda = u. \tag{3.19}$$

3.4.1.3 Complexité et implémentation

À partir de cette dernière proposition, il est possible de déduire un algorithme qui calcule une base de $H^1_{MW,c}(V, \pi_* A_K^\dagger)$ dont la proposition suivante évalue la complexité.

Proposition 3. Soient P_1 et P_2 des entiers positifs. Il existe un algorithme pour calculer une base de $H^1_{MW,c}(V, \pi_* A_K^\dagger)$ à précision analytique P_1 et précision p-adique P_2 dont le temps d'exécution est borné par $\tilde{O}(g^2 \log(q) P_1 P_2)$. Sa consommation mémoire est de $O(g \log(q) P_1 P_2)$.

De même dans [22], nous décrivons un algorithme pour calculer l'action du morphisme de Frobenius sur $H^1_{MW,c}(V, \pi_* A_K^\dagger)$. Sa complexité est donnée par la

Proposition 4. Soit C_k une courbe hyperelliptique de genre g définie sur un corps fini k de cardinalité q. Nous supposons que les points de ramification de C_k sont rationnels. Il existe un algorithme pour calculer l'action du morphisme de Frobenius sur $H^1_{MW,c}(C_k/K)$ à précision analytique P_1 et précision p-adique P_2 ayant une complexité de $\tilde{O}(g^2 \log(q) P_1 P_2) + \tilde{O}(g^3 \log(q) P_2)$ en temps et de $O(g \log(q) P_1 P_2 + g^2 \log(q) P_2)$ en mémoire.

Nous pouvons alors utiliser les deux résultats précédent afin d'obtenir un algorithme permettant de calculer le nombre de points rationnels d'une courbe hyperelliptique définie sur un corps fini de caractéristique p et cardinalité $q = p^n$.

Tout d'abord, nous devons évaluer la précision analytique P_1 et p-adique P_2 nécessaire pour les calculs. L'hypothèse de Riemann pour les courbe, nous donne qu'il est suffisant de calculer les coefficients de la matrice donnant la représentation du morphisme de Frobenius sur $H^1_{MW,c}(V, \pi_* A_K^\dagger)$ (en mettant de coté les éléments invariants de cette base) à précision $g/2.n + (2g+1)\log_p(2)$.

D'après [22, Th 2.], on peut prendre $P_1 = O(P_2)$ et nous obtenons le

Théorème 11. Soit C_k une courbe hyperelliptique de genre g sur le corps fini k. Soit n le degré absolu de k. Nous supposons que les points de ramification de C_k sont rationnels. Il existe un algorithme pour calculer le polynôme caractéristique du morphisme de Frobenius agissant sur $H^1_{MW,c}(C_k/K)$ ayant pour complexité $\tilde{O}(g^4 n^3)$ en temps et $O(g^3 n^3)$ en mémoire.

L'algorithme décrit dans les sections précédentes a été implémenté en magma [20].

3.4.1.4 Conclusion

Dans cette section, nous avons décrit un algorithme pour calculer le nombre de points rationnels d'une courbe hyperelliptique définie sur un corps fini de caractéristique impaire utilisant la cohomologie de Monsky-Washnitzer à support compact. La complexité en temps que nous obtenons est quasi-cubique en le degré absolu du corps de base. Nous remarquons que le calcul de la base peut être adapté pour des familles de courbes plus générales. La raison pour laquelle nous nous focalisons sur le cas des courbes hyperelliptiques est que pour considérer des familles plus générales il serait nécessaire de calculer des bornes logarithmiques pour les éléments d'une base de la cohomologie. Le résultat que nous avons utilisé garantit de telles bornes du moment que la matrice de connexion a des pôles simples et est préparée. Cette dernière condition sur les exposants signifie qu'ils sont des rationnels non entier ou nuls. Dans notre cas, les exposants sont 0 et $-1/2$.

Chapitre 4

Quelques calculs avec les surfaces abéliennes

4.1 La méthode CM en caractéristique 3

Dans cette section, nous résumons les résultats de [19] écrit en commun avec Robert Carls et David Kohel.

4.1.1 Équations modulaires de degré 3 et niveau 4

Dans cette section, nous donnons des équations qui ont pour solution les thêta constantes des relevés canoniques de variétés abéliennes ordinaires définies sur un corps parfait de caractéristique 3. Ces équations forment un ingrédient essentiel de l'algorithme de construction CM 3-adique décrit en section 4.1.2. L'ensemble d'équations que nous donnons est "complet" dans le sens qu'il définit un analogue en dimension plus grande de la classique courbe modulaire $X_0(3)$.

4.1.1.1 Thêta constantes du relevé canonique 3-adique

Soit R un anneau local noetherien de corps résiduel k parfait de caractéristique 3. Supposons qu'il existe $\sigma \in \mathrm{Aut}(R)$ qui relève l'automorphisme de Frobenius à la puissance 3 de k. Soit A un schéma abélien de dimension relative g sur R qui est supposé avoir une réduction ordinaire et soit $\mathscr{L}$ un fibré en droite symétrique et ample de degré 1 sur A. Nous posons $Z_n = (\mathbb{Z}/n\mathbb{Z})^g_R$ pour tout entier $n \geq 1$. Nous supposons donnée une thêta structure symétrique Θ_4 de type Z_4 pour la paire $(A, \mathscr{L}^4)$. Nous notons $(a_u)_{u \in Z_4}$ les thêta constantes pour la thêta structure θ_4. Par la suite, nous identifions Z_2 avec son image dans Z_4 via l'application $j \mapsto 2j$. Nous définissons

$$S = \{(x, y, z) \in Z_4^3 \mid (x - 2y, x + y - z, x + y + z) \in Z_2^3\}.$$

Pour $(x_1, y_1, z_1), (x_2, y_2, z_2) \in S$, nous écrivons $(x_1, y_1, z_1) \sim (x_2, y_2, z_2)$ s'il existe une matrice de permutation $P \in \mathrm{Mat}_3(\mathbb{Z})$ telle que

$$(x_1 - 2y_1, x_1 + y_1 - z_1, x_1 + y_1 + z_1) = (x_2 - 2y_2, x_2 + y_2 - z_2, x_2 + y_2 + z_2)P.$$

Théorème 12. Supposons que A soit le relevé canonique de A_k. Pour $(x,y_1,z_1),(x,y_2,z_2) \in S$ tels que $(x,y_1,z_1) \sim (x,y_2,z_2)$ on a

$$\sum_{u\in Z_2} a^{\sigma}_{y_1+u}a_{z_1+u} = \sum_{v\in Z_2} a^{\sigma}_{y_2+v}a_{z_2+v}.$$

Nous rappelons brièvement les équations modulaires de Riemann de niveau 4. Soit

$$S' = \{(v,w,x,y) \in Z_4^4 \mid (v+w, v-w, x+y, x-y) \in Z_2^4\}.$$

Pour $(v_1,w_1,x_1,y_1),(v_2,w_2,x_2,y_2) \in S'$, nous écrivons $(v_1,w_1,x_1,y_1) \sim (v_2,w_2,x_2,y_2)$ s'il existe une matrice de permutation $P \in \mathrm{Mat}_4(\mathbb{Z})$ telle que

$$(v_1+w_1, v_1-w_1, x_1+y_1, x_1-y_1) = (v_2+w_2, v_2-w_2, x_2+y_2, x_2-y_2)P.$$

Théorème 13. Pour $(v_1,w_1,x_1,y_1),(v_2,w_2,x_2,y_2) \in S'$ tels que $(v_1,w_1,x_1,y_1) \sim (v_2,w_2,x_2,y_2)$, nous avons l'égalité suivante

$$\sum_{t\in Z_2} a_{v_1+t}a_{w_1+t}\sum_{s\in Z_2} a_{x_1+s}a_{y_1+s} = \sum_{t\in Z_2} a_{v_2+t}a_{w_2+t}\sum_{s\in Z_2} a_{x_2+s}a_{y_2+s}.$$

4.1.1.2 Thêta constantes en dimension 1 et 2

Dans cette section, nous explicitons les équations du théorème 12 et du théorème 13 dans le cas de la dimension 1 et 2. Soit $\mathbb{F}_q$ un corps fini de caractéristique 3 ayant q éléments et soit $R = W(\mathbb{F}_q)$ l'anneau des vecteurs de Witt à coefficients dans $\mathbb{F}_q$. Il existe un relevé canonique $\sigma \in \mathrm{Aut}(R)$ du morphisme de Frobenius à la puissance 3 de $\mathbb{F}_q$. Soit A un schéma abélien sur R muni d'un fibré ample et symétrique $\mathscr{L}$ de degré 1 sur A.

Dimension 1. Supposons que A soit une courbe elliptique propre et lisse sur R et soient $(a_0 : a_1 : a_2 : a_3)$ les thêta constantes pour une thêta structure symétrique de type $(\mathbb{Z}/4\mathbb{Z})_R$ de $(A,\mathscr{L}^4)$. Par symétrie nous avons $a_1 = a_3$ et le théorème 13 implique que le point projectif $(a_0 : a_1 : a_2)$ appartient à une courbe lisse de genre 3 $\mathcal{A}_1(\Theta_4) \subseteq \mathrm{Proj}(\mathbb{Z}[\frac{1}{2}, x_0, x_1, x_2]) = \mathbb{P}^2_{\mathbb{Z}[\frac{1}{2}]}$ d'équation

$$(x_0^2 + x_2^2)x_0x_2 = 2x_1^4. \tag{4.1}$$

Cette dernière relation est classiquement connue comme *la relation de Riemann.* Nous remarquons que les points sur $\mathcal{A}_1(\Theta_4)$ donnent l'espace des module des courbes elliptiques munies d'une thêta structure symétrique de niveau 4.

Si nous supposons que A est a réduction ordinaire et que A est le relevé canonique de $A_{\mathbb{F}_q}$, le théorème 12 implique que les coordonnées du point projectif $(a_0 : a_1 : a_2)$ vérifient l'équation

$$x_0y_2 + x_2y_0 = 2x_1y_1, \tag{4.2}$$

où $x_i = a_i$ et $y_i = a_i^{\sigma}$ pour $i = 0,1,2$.

Dimension 2. Nous supposons maintenant que A est de dimension relative 2 sur R et que nous est donnée une thêta structure symétrique de type $(\mathbb{Z}/4\mathbb{Z})^2$ pour la paire $(A, \mathscr{L}^4)$. Soient $(a_{ij})_{(i,j)\in(\mathbb{Z}/4\mathbb{Z})^2}$ les thêta constantes pour cette dernière structure. Les relations de symétries donnent

$$a_{11} = a_{33},\ \ a_{10} = a_{30},\ \ a_{01} = a_{03},\ \ a_{13} = a_{31},\ \ a_{32} = a_{12},\ \ a_{21} = a_{23}.$$

Les *équations de Riemann* en dimension 2 sont données par

$$\begin{gathered}
(x_{00}^2 + x_{02}^2 + x_{20}^2 + x_{22}^2)(x_{00}x_{02} + x_{20}x_{22}) = 2(x_{01}^2 + x_{21}^2)^2 \\
(x_{00}^2 + x_{02}^2 + x_{20}^2 + x_{22}^2)(x_{00}x_{20} + x_{02}x_{22}) = 2(x_{10}^2 + x_{12}^2)^2 \\
(x_{00}^2 + x_{02}^2 + x_{20}^2 + x_{22}^2)(x_{00}x_{22} + x_{20}x_{02}) = 2(x_{11}^2 + x_{13}^2)^2 \\
(x_{00}x_{20} + x_{02}x_{22})(x_{00}x_{22} + x_{02}x_{20}) = 4x_{01}^2x_{21}^2 \\
(x_{00}x_{02} + x_{20}x_{22})(x_{00}x_{22} + x_{02}x_{20}) = 4x_{10}^2x_{12}^2 \\
(x_{00}x_{02} + x_{20}x_{22})(x_{00}x_{20} + x_{02}x_{22}) = 4x_{11}^2x_{13}^2 \\
(x_{00}^2 + x_{02}^2 + x_{20}^2 + x_{22}^2)x_{13}x_{11} = (x_{12}^2 + x_{10}^2)(x_{01}^2 + x_{21}^2) \\
(x_{00}^2 + x_{02}^2 + x_{20}^2 + x_{22}^2)x_{01}x_{21} = (x_{12}^2 + x_{10}^2)(x_{11}^2 + x_{13}^2) \\
(x_{00}^2 + x_{02}^2 + x_{20}^2 + x_{22}^2)x_{10}x_{12} = (x_{01}^2 + x_{21}^2)(x_{11}^2 + x_{13}^2) \\
(x_{02}x_{20} + x_{00}x_{22})x_{11}x_{13} = 2x_{01}x_{10}x_{21}x_{12} \\
(x_{20}x_{00} + x_{22}x_{02})x_{10}x_{12} = 2x_{11}x_{13}x_{21}x_{01} \\
(x_{00}x_{02} + x_{20}x_{22})x_{21}x_{01} = 2x_{11}x_{13}x_{10}x_{12}
\end{gathered} \tag{4.3}$$

$$\begin{gathered}
(x_{02}x_{20} + x_{00}x_{22})(x_{01}^2 + x_{21}^2) = 2x_{10}x_{12}(x_{11}^2 + x_{13}^2) \\
(x_{00}x_{02} + x_{20}x_{22})(x_{11}^2 + x_{13}^2) = 2x_{10}x_{12}(x_{01}^2 + x_{21}^2) \\
(x_{02}x_{20} + x_{00}x_{22})(x_{10}^2 + x_{12}^2) = 2x_{21}x_{01}(x_{11}^2 + x_{13}^2) \\
(x_{20}x_{00} + x_{22}x_{02})(x_{13}^2 + x_{11}^2) = 2x_{21}x_{01}(x_{10}^2 + x_{12}^2) \\
(x_{20}x_{00} + x_{22}x_{02})(x_{21}^2 + x_{01}^2) = 2x_{11}x_{13}(x_{10}^2 + x_{12}^2) \\
(x_{00}x_{02} + x_{20}x_{22})(x_{12}^2 + x_{10}^2) = 2x_{11}x_{13}(x_{01}^2 + x_{21}^2) \\
x_{01}x_{21}(x_{01}^2 + x_{21}^2) = x_{10}x_{12}(x_{10}^2 + x_{12}^2) \\
x_{01}x_{21}(x_{01}^2 + x_{21}^2) = x_{11}x_{13}(x_{11}^2 + x_{13}^2).
\end{gathered} \tag{4.4}$$

Par le théorème 13, le point $(a_{ij})_{(i,j)\in(\mathbb{Z}/4\mathbb{Z})^2}$ est une solution des équations (4.3) et (4.4), i.e. les équations ci-dessus sont vérifiées pour $x_{ij} = a_{ij}$. Ces équations déterminent un sous-schéma de dimension 3, $\mathcal{A}_2(\Theta_4)$, de l'espace projectif

$$\mathbb{P}^9_{\mathbb{Z}[\frac{1}{2}]} = \operatorname{Proj}\big(\mathbb{Z}[\frac{1}{2}, x_{00}, x_{01}, x_{02}, x_{10}, x_{11}, x_{12}, x_{13}, x_{20}, x_{21}, x_{22}]\big).$$

Les points sur $\mathcal{A}_2(\Theta_4)$ donnent l'espace des modules de surfaces abéliennes munies d'une thêta structure symétrique de type $(\mathbb{Z}/4\mathbb{Z})^2$. Nous remarquons que le point

$$(a_{00} : a_{01} : a_{02} : a_{10} : a_{11} : a_{12} : a_{13} : a_{20} : a_{21} : a_{22}) \in \mathbb{P}^9_{\mathbb{Z}[\frac{1}{2}]}(R)$$

est une solution des équations (4.3) et (4.4) si et seulement si les coordonnées projectives

$$\begin{gathered}
\big(a_{00}^2 + a_{02}^2 + a_{20}^2 + a_{22}^2 : 2(a_{01}^2 + a_{21}^2) : 2(a_{12}^2 + a_{10}^2) : 2(a_{11}^2 + a_{13}^2)\big), \\
\big(a_{01}^2 + a_{21}^2 : a_{00}a_{02} + a_{20}a_{22} : 2a_{11}a_{13} : 2a_{10}a_{12}\big), \\
\big(a_{12}^2 + a_{10}^2 : 2a_{11}a_{13} : a_{00}a_{20} + a_{02}a_{22} : 2a_{01}a_{21}\big), \\
\big(a_{11}^2 + a_{13}^2 : 2a_{10}a_{12} : 2a_{01}a_{21} : a_{00}a_{22} + a_{02}a_{20}\big)
\end{gathered}$$

décrivent le même point dans $\mathbb{P}^3_{\mathbb{Z}[\frac{1}{2}]}(R)$. En fait, les formules ci-dessus définissent un morphisme vers l'espace des variétés abéliennes munies d'une 2-thêta structure plongée dans $\mathbb{P}^3_{\mathbb{Z}[\frac{1}{2}]}$. Avec les équations de Riemann, le corollaire suivant du théorème 12 est à la base de notre méthode de construction CM pour les surfaces abéliennes.

Corollaire 4. Supposons que A est à réduction ordinaire et que A est le relevé canonique de $A_{\mathbb{F}_q}$. Soit (a_{ij}) les thêta constantes de A pour la 4-thêta structure symétrique. Les coordonnées du point

$$(a_{00} : a_{01} : a_{02} : a_{10} : a_{11} : a_{12} : a_{13} : a_{20} : a_{21} : a_{22}) \in \mathbb{P}^9_{\mathbb{Z}[\frac{1}{2}]}(R)$$

vérifient les relations suivantes

$$\begin{aligned} x_{00}y_{02} + x_{02}y_{00} + x_{20}y_{22} + x_{22}y_{20} - 2(x_{01}y_{01} + x_{21}y_{21}) = 0 \\ x_{00}y_{20} + x_{20}y_{00} + x_{02}y_{22} + x_{22}y_{02} - 2(x_{10}y_{10} + x_{12}y_{12}) = 0 \\ x_{00}y_{22} + x_{22}y_{00} + x_{02}y_{20} + x_{20}y_{02} - 2(x_{13}y_{13} + x_{11}y_{11}) = 0 \\ x_{01}y_{21} + x_{21}y_{01} - (x_{12}y_{10} + x_{10}y_{12}) = 0 \\ x_{01}y_{21} + x_{21}y_{01} - (x_{11}y_{13} + x_{13}y_{11}) = 0, \end{aligned} \tag{4.5}$$

où $x_{ij} = a_{ij}$ et $y_{ij} = a_{ij}^{\sigma}$.

4.1.2 Construction CM explicite en caractéristique 3

Dans cette section, nous appliquons le corollaire 4 pour la construction d'invariants CM d'une surface abélienne ordinaire en effectuant un relevé canonique en caractéristique 3. L'algorithme de construction CM se compose de deux phases distinctes
- Le calcul d'un relevé canonique grâce à un algorithme de Newton multivarié [79] utilisant les équations du corollaire 4.
- Une phase utilisant l'algorithme LLL pour reconstruire le polynôme de définition sur $\mathbb{Z}$ de l'idéal des relations entre les coordonnées relevées, suivant Gaudry et al. [42].

L'existence de l'algorithme de relèvement est une conséquence des faits suivants. Le lieu ordinaire en le premier 3 de l'espace des modules des variétés abéliennes munies d'une 4-thêta structure symétrique, qui est construit dans [96], est lisse. L'espace des paires de variétés abéliennes ordinaires munies d'une 4-thêta structure admettant une isogénie compatible de degré 3^g où g est la dimension forme un revêtement étale de ce dernier espace.

L'algorithme de relèvement s'applique à un espace des modules paramétrisé rationnellement X au dessus de $\mathbb{Z}_q$, et une intersection complète dans $X \times X$. Nous remplaçons la paramétrisation rationnelle par une paramétrisation locale analytique.

4.1.2.1 Un algorithme de relèvement pour l'espace des modules des surfaces abéliennes

Nous utilisons les notations introduites dans la section 3.0.6.1. Nous supposons que A est un schéma abélien ayant réduction ordinaire. Nous supposons que A est le relevé canonique de $A_{\mathbb{F}_q}$. Soit $\mathscr{L}$ un fibré ample symétrique de degré 1 sur A et supposons donnée une thêta structure de type $(\mathbb{Z}/4\mathbb{Z})^2$ pour $(A, \mathscr{L}^4)$. Nous notons les thêta constantes pour cette dernière thêta structure (a_{ij}) où $(i, j) \in (\mathbb{Z}/4\mathbb{Z})^2$.

Théorème 14. Il existe un algorithme déterministe qui accepte en entrée les thêta constantes $(\bar{a}_{ij})$ de $A_{\mathbb{F}_q}$ et donne en sortie les thêta constantes (a_{ij}) de A à précision $m \geq 1$, ayant pour complexité en temps

$$O(\log(m) d^{\mu} m^{\mu})$$

où $d = \log(q)$.

4.1.2.2 Reconstruction LLL

D'après la théorie de la multiplication complexe, nous savons que les invariants des relevés canoniques sont algébriques sur $\mathbb{Q}$. Nous rappelons brièvement la méthode de Gaudry et al. [42] pour retrouver via LLL les relations algébriques sur $\mathbb{Z}$. Soit γ un entier p-adique dans une extension de degré r de $\mathbb{Z}_p$, et soit m la précision à laquelle il est déterminé. Nous supposons que le degré n de son polynôme minimal sur $\mathbb{Q}$ est connu, i.e. qu'il existe $f(x) \in \mathbb{Z}[x]$, avec

$$f(\gamma) = a_n \gamma^n + \ldots + a_0 = 0,$$

où les $a_i \in \mathbb{Z}$ sont inconnus. Nous déterminons une base du noyau à gauche dans $\mathbb{Z}^{n+r+1}$ de la matrice obtenue en juxtaposant

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ \gamma_{1,0} & \gamma_{1,1} & \cdots & \gamma_{1,(r-1)} \\ \vdots & & & \vdots \\ \gamma_{n,0} & \gamma_{n,1} & \cdots & \gamma_{n,(r-1)} \end{bmatrix}$$

avec p^m fois la matrice identité de dimension r, où $\gamma_{i,j}$ sont définis par

$$\gamma^i = \gamma_{i,0} + \gamma_{i,1} w_1 + \ldots + \gamma_{i,(r-1)} w_{r-1},$$

dans les termes d'une $\mathbb{Z}_p$-base $\{1, w_1, \ldots, w_{r-1}\}$ de $\mathbb{Z}_q$. Le polynôme minimal $f(x)$ est déterminé grâce à l'algorithme LLL comme un vecteur court de $(a_0, \ldots, a_n, \varepsilon_1, \ldots, \varepsilon_r)$ dans le noyau.

La complexité de l'étape LLL dépend des valeurs de r, n et m. Les valeurs de r et n peuvent se retrouver par une sélection d'une courbe convenable et une analyse du groupe de Galois du corps de classe. La précision requise m déterminée par la taille de la sortie est bien moins comprise et nous exprimons la complexité dans les termes de ces trois paramètres. En utilisant la variante L^2 de LLL par Nguyên et Stehlé [101], l'estimation de complexité de [42] donne $O((n + r)^5 (n + r + m)m)$ en général, et dans notre cas la structure spécifique du réseau donne une complexité de $O((n + r)^4 (n + r + m)m)$.

4.1.3 Exemples de relevés canoniques

Dans cette section, nous donnons quelques exemples de relevés canoniques des thêta constantes 3-adiques. Les exemples ont été calculés en utilisant une implémentation de notre algorithme en Magma [11]. Des algorithmes génériques et une base de données des invariants CM pour les courbes de genre 2 peuvent être trouvé sur le site de D. Kohel.

Exemple 1. Considérons la courbe hyperelliptique de genre 2, $\bar{H}$ sur $\mathbb{F}_3$ définie par l'équation

$$y^2 = x^5 + x^3 + x + 1.$$

Soit $\bar{J}$ la jacobienne de $\bar{H}$. La surface abélienne $\bar{J}$ est ordinaire. Sur une extension de degré 40, il existe une thêta structure de type $(\mathbb{Z}/4\mathbb{Z})^2$ pour $(\bar{J}, \mathscr{L}^4)$ où $\mathscr{L}$ est le fibré en droite correspondant à la polarisation canonique. Soient $(\bar{a}_{ij})$ les thêta constantes de $(\bar{J}, \mathscr{L}^4)$ pour cette dernière thêta structure. On peut supposer que $\bar{a}_{00} = 1$. Notons que les coordonnées $\bar{a}_{02}$, $\bar{a}_{20}$ et $\bar{a}_{22}$ sont définies sur une extension de degré 10. Nous posons $\mathbb{F}_{3^{10}} = \mathbb{F}_3[z]$ où $z^{10} + 2z^6 + 2z^5 + 2z^4 + z + 2 = 0$. Nous choisissons

$$\bar{a}_{02} = z^{9089}, \quad \bar{a}_{20} = z^{18300} \quad \text{and} \quad \bar{a}_{22} = z^{8601}.$$

En utilisant l'algorithme décrit dans la section 4.1.2, nous relevons le triplet $(\bar{a}_{02}, \bar{a}_{20}, \bar{a}_{22})$ sur l'extension non ramifiée de $\mathbb{Z}_3$ de degré 10. Nous notons a_{02}, a_{20} les coordonnées relevées et a_{22}. Soit P_{ij} le polynôme minimal de a_{ij} sur $\mathbb{Q}$. Une recherche de relations algébrique avec l'algorithme LLL donne

$$\begin{aligned}
P_{02} &= x^{80} - 69x^{76} + 4911x^{72} + 20749x^{68} + 299094x^{64} - 202217x^{60} \\
&\quad +1093161x^{56} - 7393871x^{52} + 11951456x^{48} + 7541235x^{44} \\
&\quad -26349059x^{40} + 7541235x^{36} + 11951456x^{32} - 7393871x^{28} \\
&\quad +1093161x^{24} - 202217x^{20} + 299094x^{16} + 20749x^{12} + 4911x^8 \\
&\quad -69x^4 + 1, \\
P_{20} &= x^{20} - 5x^{19} + 23x^{18} - 53x^{17} + 112x^{16} - 203x^{15} + 279x^{14} - 345x^{13} \\
&\quad +360x^{12} - 333x^{11} + 329x^{10} - 333x^9 + 360x^8 - 345x^7 + 279x^6 \\
&\quad -203x^5 + 112x^4 - 53x^3 + 23x^2 - 5x + 1, \\
P_{22} &= x^{80} + 5x^{76} + 184x^{72} + 2254x^{68} + 4470x^{64} + 160109x^{60} + 768428x^{56} \\
&\quad +421488x^{52} + 36971535x^{48} - 75225290x^{44} + 44767882x^{40} \\
&\quad -43287046x^{36} + 86078086x^{32} - 75568556x^{28} + 31873762x^{24} \\
&\quad -7293064x^{20} + 989181x^{16} - 32859x^{12} + 4318x^8 + 44x^4 + 1.
\end{aligned}$$

Nous concluons que le corps k_0 engendré par les coordonnées a_{02}, a_{20} et a_{22} est une extension Galoisienne de $\mathbb{Q}$ de degré 160. Notons que k_0 contient $\mathbb{Q}(i)$.

Le polynôme caractéristique du morphisme de Frobenius absolu de $\bar{J}$ est

$$x^4 + 3x^3 + 5x^2 + x + 9.$$

Soit $K = \mathrm{End}_{\mathbb{F}_3}(\bar{J}) \otimes \mathbb{Q}$. Le corps K est un corps CM normal de dimension 4 dont le groupe de Galois est égal à $\mathbb{Z}/4\mathbb{Z}$. Le nombre de classes de K est égal à 1. Le sous-corps réel maximal de K est donné par $\mathbb{Q}(\sqrt{13})$. Notons que K est égal à son propre corps reflex K^*. Le compositum k_0K^* forme une extension abélienne de K^* ayant pour conducteur 8 et groupe de Galois $(\mathbb{Z}/2\mathbb{Z})^2 \times (\mathbb{Z}/10\mathbb{Z})$. Notons que le polynôme P_{20} engendre le corps de classe de rayon de K^* modulo 2.

Nous remarquons que la courbe H ayant pour équation

$$y^2 = 52x^5 - 156x^4 + 208x^3 - 156x^2 + 64x - 11$$

est un relevé canonique de $\bar{H}$ dans le sens que H se réduit sur $\bar{H}$ et que la jacobienne de H est isomorphe au relevé canonique de $\bar{J}$. Pour une liste de courbes de genre 2 ayant multiplication complexe nous revoyons le lecteur vers [123].

Exemple 2. Soit $\bar{H}$ la courbe hyperelliptique sur $\mathbb{F}_{3^6} = \mathbb{F}_3[z]$ avec $z^3 - z + 1 = 0$, définie par l'équation affine

$$y^2 = x(x-1)(x-z)(x-z^8)(x-z^2).$$

Nous pouvons associer des thêta constantes à $\bar{H}$ sur une extension et appliquer notre algorithme afin de déterminer le relevé canonique des invariants de Rosenhain à partir de la liste des thêta constantes. En utilisant l'algorithme LLL, les invariants de Igusa

$$j_1 = \frac{J_2^5}{J_{10}}, \quad j_2 = \frac{J_2^3 J_4}{J_{10}}, \quad j_4 = \frac{J_2 J_8}{J_{10}},$$

de la courbe relevée canoniquement H satisfont la relation

$$\begin{aligned}
&116757924411252876637960400005285561864702968 3j_1^6 \\
&\quad - 152576778498036139555712362221331427936276660398901315481108 48j_1^5 \\
&\quad + 119613187927709421321323782662565661666729098621643912069623876959810355 2j_1^4 \\
&\quad - 15026901839645385662905995514419940545045030890784639316486791370893169241620 48j_1^3 \\
&\quad + 94949600514980451348563662445123861714429688472687490461831537573194959983476736 00000j_1^2 \\
&\quad - 94892424945327681986219937537595326692680634607252685632729203963434893855581798 40000000000j_1 \\
&\quad + 31547451835582324333091829027216544894002126520451921018745800900736827133337272 32000000000000000 \\
&31524639591038276692249308001427101703469801441j_2^6 \\
&\quad - 16745634807723620828207592940844036495138204085628428409110528j_2^5 \\
&\quad - 12265164179615739710029144012197055859859725320474999182497036825001984j_2^4 \\
&\quad + 35214177531903280346028564046053042847680522703280784178837536706828592742 4j_2^3 \\
&\quad - 11588611701570137317081804138762727639770955607998908195477045771454843453440000 0j_2^2 \\
&\quad + 62410881010002047470123155597613207866129246215906414112791308963958017228800000 00000j_2 \\
&\quad - 11911694866700746148345021028981415501863609754427784385834331978459198259200000 000000000000 \\
&229814622618669037086497455330403571418294852504 89j_4^6 \\
&\quad - 38333133385822330975872342595626396239705000243787196311246336j_4^5 \\
&\quad - 13445890564402694049486311582599736771794395285600128293985309687808j_4^4 \\
&\quad - 255870832830872991577269047893520950236274153918508961754273160951234 56j_4^3 \\
&\quad - 20922653078662308982945894934868322119306736601817862795598824527101952000j_4^2 \\
&\quad - 61259814230097056731768967829978518304429009163243510825472679508705280000 00j_4 \\
&\quad - 12260055755474262524570670484641566489377734821667749961858456108400640000000000
\end{aligned}$$

Nous notons que aucune de ces jacobiennes n'ont bonne réduction ordinaire en 2 et donc étendent le domaine d'application de la méthode CM 2-adique de [42].

4.1.4 Conclusion

Ce travail généralise des travaux antérieurs sur les relevés canoniques 2-adiques en dimension supérieure dans le cas 3-adique. Nous avons tout d'abord développé un analogue de la courbe modulaire $X_0(3)$. Ensuite, nous avons décrit un algorithme de relèvement de Hensel dans le cadre analytique (enlevant l'hypothèse d'une paramétrisation rationnelle de la variété). Comme application de notre travail, nous avons obtenu des constructions CM explicites de modules de courbes de genre 2 (et de leurs surfaces jacobiennes) donnant une alternative 3-adique à la construction 2-adique de Gaudry et al. [42], et de ce fait nous avons étendu le domaine d'application de cette méthode à d'autres corps CM de degré 4.

4.2 Surfaces de Kummer en caractéristique 2

Dans cette section, nous résumons les résultats de [43] écrit en commun avec Pierrick Gaudry.

4.2.1 Équation en caractéristique 2

Soit k un corps fini de caractéristique 2. Dans ce paragraphe, nous donnons l'équation générale d'une surface de Kummer ordinaire définie sur k.

Nous montrons dans [43] que la donnée d'une quartique de Kummer est la même chose que la donnée d'un triplet $(A_k, \mathscr{L}_k, \Theta_\delta)$ où k est une extension finie quelconque de $\mathbb{F}_2$, A_k est variété abélienne ordinaire sur k, $\mathscr{L}_k$ est un fibré en droite ample totalement symétrique de degré 2 et Θ_δ est une thêta structure de type $\delta = (2,2)$ définie sur k. La proposition suivante classifie ces dernières structures (à comparer avec [60] proposition 4.1)

Proposition 5. Soit $\delta = (2,2)$. Il existe une correspondance bijective entre
- l'ensemble des triplets $(A_k, \mathscr{L}_k, \Theta_\delta)$ où k est une extension finie quelconque de $\mathbb{F}_2$, A_k est une variété abélienne ordinaire sur k, $\mathscr{L}_k$ est un fibré en droite ample totalement symétrique de degré 2 et Θ_δ est une thêta structure de type δ définie sur k,
- et l'ensemble des triplets d'éléments $(b', c', d') \in k^3$ tels que $b'c'd' \neq 0$.

Soit $(b', c', d') \in k^3$, une équation pour la surface de Kummer $K_{(1:b':c':d')}$ est donnée par

$$b'c'd'XYZT + c'^2b'^2(X^2T^2 + Y^2Z^2) \\ + b'^2d'^2(X^2Z^2 + Y^2T^2) + c'^2d'^2(X^2Y^2 + T^2Z^2) = 0.$$

4.2.2 Formules pour la loi de pseudo-groupe

Soit k une extension finie de $\mathbb{F}_2$ et soit $\overline{k}$ la clôture algébrique de k. Soit $(b', c', d') \in k^3$ tel que $b'c'd' \neq 0$ et soit $K_{(1:b':c':d')}$ la surface de Kummer associée à (b', c', d') par la proposition 5. Soit A_k la variété abélienne telle que son quotient par l'action du morphisme inverse i donne $K_{(1:b':c':d')}$ et notons $\pi : A_k \to K_{(1:b':c':d')}$ la projection naturelle. Dans toute cette section, si P est un point géométrique de A_k, nous notons par $\overline{P}$ le point $\pi(P)$. De la même manière, la notation $\overline{P}$ signifie que $\overline{P}$ est un point géométrique de $K_{(1:b':c':d')}$ représenté par un point P sur A_k.

Nous donnons un algorithme afin de calculer la loi de pseudo-groupe sur $K_{(1:b':c':d')}$.

Algorithme de doublement : `DoubleKummer(`$\overline{P}$`)`
Entrée : $\overline{P} = (x : y : z : t)$ un $\overline{k}$-point de $K_{(1:b':c':d')}$;
Sortie : Le doublement $2\overline{P} = (x' : y' : z' : t')$ dans $K_{(1:b':c':d')}(\overline{k})$.

1. $x' = (x^2 + y^2 + z^2 + t^2)^2$;
2. $y' = \frac{1}{b'}(xy + zt)^2$;
3. $z' = \frac{1}{c'}(xz + yt)^2$;
4. $t' = \frac{1}{d'}(xt + yz)^2$;
5. Retourne $2\overline{P} = (x', y', z', t')$.

Algorithme de pseudo-addition : `PseudoAddKummer(`$\overline{P}$`, `$\overline{Q}$`, `$\overline{R}$`)`
Entrée : $\overline{P} = (x : y : z : t)$ et $\overline{Q} = (\underline{x} : \underline{y} : \underline{z} : \underline{t})$ deux $\overline{k}$-points de $K_{(1:b':c':d')}$ et $\overline{R} = (\bar{x} : \bar{y} : \bar{z} : \bar{t})$ qui est soit $\pi(P + Q)$ ou $\pi(P - Q)$, avec $\bar{x}\bar{y}\bar{z}\bar{t} \neq 0$.
Sortie : Le point $(x' : y' : z' : t')$ égal à $\pi(P + Q)$ ou $\pi(P - Q)$ qui est différent de $\overline{R}$.

1. $x' = (x\underline{x} + y\underline{y} + z\underline{z} + t\underline{t})^2/\bar{x}$;
2. $y' = (x\underline{y} + y\underline{x} + z\underline{t} + t\underline{z})^2/\bar{y}$;
3. $z' = (x\underline{z} + z\underline{x} + y\underline{t} + t\underline{y})^2/\bar{z}$;
4. $t' = (x\underline{t} + t\underline{x} + y\underline{z} + z\underline{y})^2/\bar{t}$;
5. Retourne $(x', y', z', t') = \pi(P + Q)$ ou $\pi(P - Q)$.

Le calcul du doublement nécessite 6 multiplications, 3 multiplications par une constante qui ne dépend que de la surface de Kummer et 5 carrés. La calcul de la pseudo-addition nécessite 16 multiplications, 4 carrés et 4 divisions.

Ces décomptes peuvent être réduits en utilisant le fait que dans le cadre de l'échelle de Montgomery, le point base est toujours le même dans la pseudo-addition et donc les 4 divisions peuvent être remplacées par 3 multiplications. Pour le doublement, on peut par exemple calculer les 4 produits xt, yz, $(x+y)(z+t)$ et $(x+z)(y+t)$, à partir de quoi on déduit $xt+yz$, $xz+yt = (x+y)(z+t)+xt+yz$ et $xy+zt = (x+z)(y+t)+xt+yz$. Donc le coût du doublement est de 4 multiplications, 3 multiplications par une constante, et 5 carrés. De manière similaire la pseudo-addition peut se faire de la manière suivante. Posons

$$\begin{array}{rcl}
L_1 &=& x\underline{x} \\
L_2 &=& y\underline{y} \\
L_3 &=& z\underline{z} \\
L_4 &=& t\underline{t} \\
M_1 &=& (y + z + t)(\underline{y} + \underline{z} + \underline{t}) \\
M_2 &=& (x + y + t)(\underline{x} + \underline{y} + \underline{t}) \\
M_3 &=& (x + z + t)(\underline{x} + \underline{z} + \underline{t}) \\
N &=& (x + y + z + t)(\underline{x} + \underline{y} + \underline{z} + \underline{t})
\end{array}$$

Alors $L_1+L_2+L_3+L_4$, $L_3+L_4+M_1+M_3+N$, $L_2+L_4+M_1+M_2+N$, $L_1+L_4+M_2+M_3+N$ donnent le résultat pour la pseudo-addition. Ainsi, la pseudo-addition coûte 11 multiplications et 4 carrés. Finalement, dans une échelle de Montgomery, il faut faire un doublement et une pseudo-addition pour chaque bit de scalaire et nous obtenons le résultat suivant

Théorème 15. Multiplier par un scalaire un point sur une surface de Kummer de coordonnée non nulle coûte 9 carrés, 15 multiplications générales et 3 multiplications par une constante qui ne dépend que de la surface de Kummer pour chaque bit de scalaire.

4.2.3 Le cas du genre 1

Toutes les lignes de Kummer définies sur k sont isomorphes comme variété algébrique à $\mathbb{P}^1_k$. Seule la loi de pseudo-groupe change. Dans cette section, nous donnons des algorithmes pour calculer la loi de pseudo-groupe. Nous distinguons le cas de la caractéristique paire et impaire.

4.2.3.1 Le cas de la caractéristique paire.

Soit k une extension finie de $\mathbb{F}_2$. Les résultats sur les surfaces de Kummer en caractéristique 2 se transposent sans problème au cas de la dimension 1. En particulier, l'ensemble des lignes de Kummer ordinaires définies sur k est paramétrisé par les points de $\mathcal{A}^1_k$ définis sur k. Soient $b' \in k$ et $K_{(1:b')}$ la ligne de Kummer associée. Nous conservons les conventions de la section précédente : soit E_k une courbe elliptique telle que son quotient par le morphisme d'inversion i_k donne $K_{(1:b')}$. Notons $\pi : E_k \to K_{(1:b')}$ la projection naturelle. La notation $\overline{P}$ signifie que $\overline{P}$ est un point géométrique de $K_{(1:b')}$ représenté par un point P de E_k.

Algorithme de doublement : `DoubleKummer(`$\overline{P}$`)`
Entrée : Un point $\overline{P} = (x : y)$ de $K_{(1:b')}(\overline{k})$;
Sortie : Le double $2\overline{P} = (x' : y')$ dans $K_{(1:b')}(\overline{k})$.

1. $x' = (x^2 + y^2)^2$;
2. $y' = \frac{1}{b'}(xy)^2$;
3. Retourne $2\overline{P} = (x' : y')$.

Algorithme de pseudo-addition : `PseudoAddKummer(`$\overline{P}$`, `$\overline{Q}$`, `$\overline{R}$`)`
Entrée : $\overline{P} = (x : y)$ et $\overline{Q} = (\underline{x} : \underline{y})$ sur $K_{(1:b')}$ et $\overline{R} = (\bar{x}, \bar{y})$ qui est soit $\pi(P+Q)$ ou $\pi(P-Q)$, avec $\bar{x}\bar{y} \neq 0$.
Sortie : Le point $(x' : y')$ égal à $\pi(P+Q)$ ou $\pi(P-Q)$ qui est différent de $\overline{R}$.

1. $x' = (x\underline{x} + y\underline{y})^2/\bar{x}$;
2. $y' = (x\underline{y} + y\underline{x})^2/\bar{y}$;
3. Retourne $(x', y') = \pi(P+Q)$ ou $\pi(P-Q)$.

Dans cette formule, nous reconnaissons la variante des formules de Lopez-Dahab donnée par Stam dans [118].

4.2.3.2 Le cas de la caractéristique impaire.

Dans cette section, nous considérons p un premier impair et k une extension finie de $\mathbb{F}_p$. Dans ce cas, comme tout a bonne réduction, nous pouvons utiliser le principe de Lefschetz pour transporter tous les résultats connus sur $\mathbb{C}$. Cela nous donne que l'ensemble des variétés de Kummer définies sur k est paramétrisé par les k-points de $\mathbb{P}^1_k$. Soit $(a : b)$ les coordonnées homogènes d'un k-point de $\mathbb{P}^1_{\mathbb{F}_p}$ qui définit une ligne de Kummer $K_{(a:b)}$. Comme variété algébrique, $K_{(a:b)}$ est isomorphe à $\mathbb{P}^1_k$. Nous conservons les mêmes notations que dans la section précédente pour E_k, π, P et $\overline{P}$. Soient $A' = (a^2+b^2)/2$ et $B' = (a^2-b^2)/2$. La loi de pseudo-groupe est donnée par les algorithmes suivants.

Algorithme de doublement : `Double(`$\overline{P}$`)`
Entrée : Un point $\overline{P} = (x : y)$ dans $K_{(a:b)}(\overline{k})$.
Sortie : Le double $2\overline{P} = (x' : y')$.

1. $x_0 = (x^2 + y^2)$;
2. $y_0 = \frac{A'}{B'}(x^2 - y^2)$;
3. $x' = (x_0 + y_0)$;
4. $y' = \frac{a}{b}(x_0 - y_0)$;
5. Retourne $(x' : y')$.

Algorithme de pseudo-addition : `PseudoAdd(`$\overline{P}$`, `$\overline{Q}$`, `$\overline{R}$`)`
Entrée : Deux points $\overline{P} = (x : y)$ et $\overline{Q} = (\underline{x} : \underline{y})$ sur $K_{(a:b)}$, et $\overline{R} = (\bar{x} : \bar{y})$ qui est soit $\pi(\overline{P} + \overline{Q})$ ou $\pi(\overline{P} - \overline{Q})$, avec $\bar{x}\bar{y} \neq 0$.
Sortie : Le point $(x' : y')$ égal à $\pi(\overline{P} + \overline{Q})$ ou $\pi(\overline{P} - \overline{Q})$ qui est différent de $\overline{R}$.

1. $x_0 = (x^2 + y^2)(\underline{x}^2 + \underline{y}^2)$;
2. $y_0 = \frac{A'}{B'}(x^2 - y^2)(\underline{x}^2 - \underline{y}^2)$;
3. $X = (x_0 + y_0)/\bar{x}$;
4. $Y = (x_0 - y_0)/\bar{y}$;
5. Retourne $(x' : y')$.

Ces formules se déduisent directement à partir de la théorie classique des fonctions thêta.

Par exemple, l'algorithme de doublement découle directement des égalités :

$$\left\{\begin{array}{rcl} a\vartheta_2\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](\mathbf{z},\tau) & = & \vartheta_2\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](\mathbf{z},2\tau)^2 + \vartheta_2\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right](\mathbf{z},2\tau)^2 \\ b\vartheta_2\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right](\mathbf{z},\tau) & = & \vartheta_2\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](\mathbf{z},2\tau)^2 - \vartheta_2\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right](\mathbf{z},2\tau)^2 \end{array}\right.$$

$$\left\{\begin{array}{rcl} 2A\vartheta_2\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](2\mathbf{z},2\tau) & = & \vartheta_2\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](\mathbf{z},\tau)^2 + \vartheta_2\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right](\mathbf{z},\tau)^2 \\ 2B\vartheta_2\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right](2\mathbf{z},2\tau) & = & \vartheta_2\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](\mathbf{z},\tau)^2 - \vartheta_2\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right](\mathbf{z},\tau)^2 \end{array}\right.$$

Dans l'algorithme de doublement et de pseudo-addition, les formules commencent par mettre au carré les coordonnées des points de départ. En conséquence, il est plus efficace de manipuler les carrés des coordonnées afin de partager le calcul de ces carrés.

De la même manière que pour la forme de Montgomery, il est possible de concevoir des chaînes d'addition qui permettent de multiplier un point sur la ligne de Kummer par un scalaire. L'échelle binaire produit le décompte d'opérations suivant pour la multiplication par un scalaire :

Théorème 16. Multiplier par un scalaire sur une ligne de Kummer avec un point de coordonnées non nulles coûte 6 carrés, 3 multiplications générales et 3 multiplications par une constante qui ne dépend que de la ligne de Kummer, cela pour chaque bit de scalaire.

4.2.4 Implémentation et résultats

Nous résumons les coûts pour les différentes formules basées sur les fonctions thêta en genre 1 et 2 en caractéristique paire et impaire. Dans cette table, M représente une multiplication générale, S un carré, et D une multiplication par une constante qui ne dépend que de la courbe ou de la surface et donc peut être considérée comme petite. Dans ces estimations, nous supposons qu'une échelle binaire est utilisée, de manière à ce que les divisions qui interviennent dans la pseudo-addition le sont par des éléments qui sont constants tout le temps de la multiplication par un scalaire.

Coût par bit d'une multiplication scalaire	
Courbe elliptique, caractéristique impaire	3 M + 6 S + 3 D
Courbe elliptique, caractéristique paire	5 M + 5 S + 1 D
Genre 2, caractéristique impaire	7 M + 12 S + 9 D
Genre 2, caractéristique paire	15 M + 9 S + 3 D

Ces formules ont été implémentées en utilisant la librairie $\texttt{mp}\mathbb{F}_q$ [46], afin d'obtenir les temps de calcul de la table qui suit. Les plate-formes de test sont un AMD Opteron 250 2.4 Ghz et un Intel Core2 Duo E6700 2.66 Ghz, sous linux en mode 64-bits. `curve25519` est le cryptosystème décrit dans [4] et utilise une courbe elliptique sous sa forme de Montgomery sur un corps premier de 255 bits. `surf127eps` utilise une surface de Kummer à multiplication complexe au dessus du corps premier de 127 bits. `curve2251` utilise une courbe elliptique définie sur $\mathbb{F}_{2^{251}}$, et `surf2113` utilise une surface de Kummer définie sur $\mathbb{F}_{2^{113}}$.

Dans le cas du genre 1 et de la caractéristique impaire, nous avons utilisé `curve25519` qui se sert de la forme de Montgomery à la place des formules présentées dans ce papier puisque le gain de n'avoir à faire qu'un carré au lieu de une multiplication ne compense par l'addition supplémentaire par une petite constante. Les autres formules utilisent les fonctions thêtas.

Le système `surf127eps` utilise une courbe CM, puisque le comptage de points est problématique pour cette taille. En conséquence, les coefficients dépendant de la surface dans les formules ressemblent à des éléments aléatoires et faire une multiplication par ces éléments n'est pas moins coûteux qu'une multiplication générique.

Temps en cycles CPU pour une multiplication scalaire				
	`curve25519`	`surf127eps`	`curve2251`	`surf2113`
Opteron K8	310,000	296,000	1,400,000	1,200,000
Core2	386,000	405,000	888,000	687,000

4.2.5 Conclusion

Les genre 3 et 4 sont susceptible d'un traitement similaire, mais du fait des progrès dans le domaine du calcul de logarithmes discrets, il parait raisonnable pour les applications cryptographiques de ne regarder que les cas des genres 1 et 2.

D'un point de vue théorique, une équation quartique pour une surface de Kummer non ordinaire est donnée dans [61]. Mais la question des formules de pseudo-addition sur de telles surfaces de Kummer non ordinaires reste ouverte. En utilisant les coordonnées de Mumford et des formules à la Cantor, la loi de groupe peut être plus efficace dans le cas non ordinaire, qui présente donc un certain intérêt.

4.3 Travaux en cours

Dans cette section, nous présentons des travaux correspondant à des articles non encore acceptés pour publications.

4.3.1 Calcul de correspondances modulaires

Dans cette section, nous résumons les résultats de [35] écrit en commun avec Jean-Charles Faugère.

4.3.1.1 Thêta constantes et isogénies

Dans cette section, nous nous intéressons à la situation suivante. Soit ℓ et n des entiers premiers entre eux. Soit $(A_k, \mathscr{L}, \Theta_{\overline{\ell n}})$ une variété abélienne de dimension g munie d'un $(\overline{\ell n})$ marquage [96]. Cela signifie que $\mathscr{L}$ est un fibré ample symétrique et que $\Theta_{\overline{\ell n}}$ est une thêta structure symétrique de type $(\overline{\ell n})$. Nous rappelons que la thêta structure $\Theta_{\overline{\ell n}}$ induit une décomposition du noyau de la polarisation

$$K(\mathscr{L}) = K_1(\mathscr{L}) \times K_2(\mathscr{L}) \tag{4.6}$$

en des sous-groupes maximaux isotropes pour le couplage donné par les commutateurs de $G(\mathscr{L})$. Soit K un sous-groupe maximal isotrope de ℓ-torsion de $K(\mathscr{L})$ compatible avec la décomposition (4.6). Il y a deux possibilités pour un choix de K, un contenu dans $K_1(\mathscr{L})$, l'autre dans $K_2(\mathscr{L})$. Dans le paragraphe suivant, nous expliquons qu'un choix de K détermine une certaine variété abélienne avec un $\overline{n}$-marquage.

Soit X_k le quotient de A_K par K et soit $\pi : A_k \to X_k$ la projection naturelle. Soit $\kappa : G(\mathscr{L}) \to K(\mathscr{L})$ la projection naturelle déduite du diagramme (3.3.1). Comme K est un sous-groupe G de $G(\mathscr{L})$ défini par $G = \kappa^{-1}(K)$. Soit $\tilde{K}$ le sous-groupe de niveau de $G(\mathscr{L})$ défini comme l'intersection de G avec l'image de $(1, x, y)_{(x,y)\in Z(\overline{\ell n})\times \hat{Z}(\overline{\ell n})} \subset \mathcal{H}(\overline{\ell n})$

par $\Theta_{\overline{\ell n}}$. Par la théorie de descente de Grothendieck, nous savons que la donnée de $\tilde{K}$ est équivalente à la donnée d'un fibré $\mathscr{X}$ sur X_k ainsi qu'un isomorphisme $\lambda : \pi^*(\mathscr{X}) \to \mathscr{L}$.

Maintenant, nous expliquons que le $(\overline{\ell n})$-marquage sur A_k induit un $\overline{n}$-marquage sur X_k. Soit $G^*(\mathscr{L})$ le centralisateur de $\tilde{K}$ dans $G(\mathscr{L})$. En appliquant, [95] Proposition 2 p. 291, nous obtenons un isomorphisme

$$G^*(\mathscr{L})/\tilde{K} \simeq G(\mathscr{X}) \tag{4.7}$$

et en conséquence de quoi une projection naturelle $q : G^*(\mathscr{L}) \to G(\mathscr{X})$.

Comme $\mathcal{H}(\overline{n}) = \mathbb{G}_m \times Z(\overline{n}) \times \hat{Z}_{\overline{n}}$, afin de définir une thêta structure $\Theta_{\overline{n}} : \mathcal{H}(\overline{n}) \to G(\mathscr{X})$, il suffit de donner des morphismes $1_{\mathbb{G}_m} \times 0_{Z(\overline{n})} \times \hat{Z}(\overline{n}) \to G(\mathscr{X})$ et $1_{\mathbb{G}_m} \times Z(\overline{n}) \times 0_{\hat{Z}(\overline{n})} \to G(\mathscr{X})$. Posons $Z^*(\overline{\ell n}) = \Theta_{\overline{\ell n}}^{-1}(G^*(\mathscr{L})) \cap Z(\overline{\ell n})$ et $\hat{Z}^*(\overline{\ell n}) = \Theta_{\overline{\ell n}}^{-1}(G^*(\mathscr{L})) \cap \hat{Z}(\overline{\ell n})$.

Par définition du couplage donné par les commutateurs, nous avons $\hat{Z}^*(\overline{\ell n}) = \hat{Z}(\overline{\ell n})$ ou $\hat{Z}^*(\overline{\ell n}) = \hat{Z}(\overline{n})$ en fonction du choix de $\tilde{K}$. En conséquence de quoi, il existe une projection naturelle $p : Z^*(\overline{\ell n}) \to Z(\overline{n})$. De la même manière, $Z^*(\overline{\ell n}) = Z(\overline{\ell n})$ or $Z^*(\overline{\ell n}) = Z(\overline{n})$ et nous avons une injection naturelle $i : Z(\overline{n}) \to Z^*(\overline{\ell n})$.

Nous pouvons définir $\Theta_{\overline{n}}$ comme l'unique thêta structure pour $\mathscr{X}$ telle que les diagrammes suivants commutent

$$\begin{tikzcd} (1,0,y)_{y \in \hat{Z}^*(\overline{\ell n})} \arrow[r, "\Theta_{\overline{\ell n}}"] \arrow[d, "p"] & G^*(\mathscr{L}) \arrow[d, "q"] \\ (1,0,y)_{y \in \hat{Z}(\overline{n})} \arrow[r, "\Theta_{\overline{n}}"] & G(\mathscr{X}) \end{tikzcd} , \tag{4.8}$$

$$\begin{tikzcd} (1,0,y)_{y \in Z^*(\overline{\ell n})} \arrow[r, "\Theta_{\overline{\ell n}}"] & G^*(\mathscr{L}) \arrow[d, "q"] \\ (1,0,y)_{y \in Z(\overline{n})} \arrow[r, "\Theta_{\overline{n}}"] \arrow[u, "i"] & G(\mathscr{X}) \end{tikzcd} , \tag{4.9}$$

où i est déduit de l'injection naturelle et p se déduit de la projection naturelle.

Nous disons que les thêta structures $\Theta_{\overline{\ell n}}$ et $\Theta_{\overline{n}}$ sont π-compatibles si les diagrammes (4.8) et (4.9) commutent.

Soient K_1 et K_2 les sous-groupes de ℓ-torsion maximaux de $K_1(\mathscr{L})$ et $K_2(\mathscr{L})$. En prenant $K = K_2$ et $K = K_1$ dans la construction précédente, nous obtenons respectivement $(B_k, \mathscr{L}_0, \Theta_{\overline{n}})$ et $(C_k, \mathscr{M}, \Theta'_{\overline{n}})$ deux variétés abéliennes munies d'un $\overline{n}$-marquage. Soit $\pi : A_k \to B_k$ et $\pi' : A_k \to C_k$ les isogénies déduites à partir de la construction. Nous disposons alors d'une correspondance modulaire bien définie

$$\Phi_\ell : \mathcal{M}_{\overline{\ell n}} \to \mathcal{M}_{\overline{n}} \times \mathcal{M}_{\overline{n}}. \tag{4.10}$$

Soit $[\ell]$ l'isogénie de multiplication par ℓ sur B_k et soit $\hat{\pi} : B_k \to A_k$ l'isogénie duale de $\pi : A_k \to B_k$. Nous avons $[\ell] = \pi \circ \hat{\pi}$. Comme $\mathscr{L}_0$ est symétrique, en appliquant la formule de [95] p. 289, nous avons $[\ell]^*\mathscr{L}_0 = \mathscr{L}_0^{\ell^2}$ et le diagramme suivant montre que

C_k est obtenue en quotientant B_k par le sous-groupe isotrope maximal de $(B_k, \mathscr{L}_0^{\ell^2})$ d'ordre ℓ^{2g}.

$$\begin{tikzcd}
B_k \arrow[dd, "{[\ell]}"'] \arrow[dr, "\hat{\pi}"] & & \\
& A_k \arrow[dl, "\pi"] \arrow[dr, "\pi'"] & \\
B_k & & C_k
\end{tikzcd}\quad .$$

Les deux propositions suivantes expliquent les relations entre les thêta constantes de $(A_k, \mathscr{L}, \Theta_{\overline{\ell n}})$ et les thêta constantes de $(B_k, \mathscr{L}_0, \Theta_{\overline{n}})$ et $(C_k, \mathscr{M}, \Theta'_{\overline{n}})$. En gardant les notations du paragraphe précédent, nous avons

Proposition 6. Soient $(A_k, \mathscr{L}, \Theta_{\overline{\ell n}})$ et $(B_k, \mathscr{L}_0, \Theta_{\overline{n}})$ définies comme au dessus. Il existe un facteur constant $\omega \in \overline{k}$ tel que pour tout $i \in Z(\overline{n})$, nous avons $\pi^*(\vartheta_i^{\Theta_{\overline{n}}}) = \omega \vartheta_i^{\Theta_{\overline{\ell n}}}$. Dans cette dernière relation, $Z(\overline{n})$ est identifié à un sous-groupe de $Z(\overline{\ell n})$ via l'application $x \mapsto \ell x$.

De cette dernière proposition, nous tirons le corollaire

Corollaire 5. Soient $(A_k, \mathscr{L}, \Theta_{\overline{\ell n}})$ et $(B_k, \mathscr{L}_0, \Theta_{\overline{n}})$ définie comme au dessus. Soient $(a_u)_{u\in Z(\overline{\ell n})}$ et $(b_u)_{u \in Z(\overline{n})}$ les thêta constantes associées respectivement à $(A_k, \mathscr{L}, \Theta_{\overline{\ell n}})$ et $(B_k, \mathscr{L}_0, \Theta_{\overline{n}})$. Nous avons pour tout $u \in Z(\overline{n})$, $b_u = a_u$.

Proposition 7. Soient $(A_k, \mathscr{L}, \Theta_{\overline{\ell n}})$ et $(C_k, \mathscr{L}_0, \Theta_{\overline{n}})$ définies comme au dessus. Soient $(a_u)_{u\in Z(\overline{\ell n})}$ et $(c_u)_{u \in Z(\overline{n})}$ les thêta constantes associées respectivement à $(A_k, \mathscr{L}, \Theta_{\overline{\ell n}})$ et $(C_k, \mathscr{L}_0, \Theta_{\overline{n}})$. Soit K' le noyau de π'. Soit Z', le sous-groupe de $K(\overline{\ell n})$ défini par $\overline{\Theta}_{\overline{\ell n}}^{-1}(K')$. Par construction, nous savons que Z' est un sous-groupe de $Z(\overline{\ell n})$. Soit $\phi : Z(\overline{n}) \to Z(\overline{\ell n})$ l'application définie par $j \mapsto \ell j$. Nous avons pour tout $u \in Z(\overline{\ell n})$,

$$c_u = \sum_{t \in Z'} a_{\phi(u)+t}. \tag{4.11}$$

4.3.1.2 L'image de la correspondance modulaire

Dans cette section, nous utilisons les résultats des sections précédentes afin de décrire par des équations l'image de la correspondance modulaire Φ_ℓ. Plus précisément, soit $(B_k, \mathscr{L}_0, \Theta_{\overline{n}})$ une variété abélienne munie d'un $\overline{n}$-marquage et notons par $(b_u)_{u\in Z(\overline{n})}$ ses thêta constantes. Soit $\mathscr{C}$ la sous-variété de $\mathcal{M}_{\overline{n}} \times \mathcal{M}_{\overline{n}}$ image de $\Phi_\ell(\mathcal{M}_{\overline{\ell n}})$ dans $\mathcal{M}_{\overline{n}} \times \mathcal{M}_{\overline{n}}$.

Notons π_1 (resp. π_2) la restriction à $\mathscr{C}$ de la première (resp. seconde) projection de $\mathcal{M}_{\overline{n}} \times \mathcal{M}_{\overline{n}}$ dans $\mathcal{M}_{\overline{n}}$. Nous voudrions calculer la variété algébrique $\pi_1^{-1}((b_u)_{u\in Z(\overline{n})})$. Nous remarquons que cette question est l'analogue dans notre situation au calcul des solutions de l'équation $\Phi_\ell(j, X)$ définie à partir du polynôme modulaire Φ_ℓ et de j un certain j-invariant.

Soit I l'idéal de l'anneau des polynômes multivariés $k[x_u|u \in Z(\overline{\ell n})]$ engendré par les relations du théorème 7, ou les coefficients sont pris dans k, ainsi que par les relations de symétrie $b_u = b_{-u}$ pour tout $u \in Z_{\overline{n}}$. Soit J l'image de I par l'application de spécialisation

$$k[x_u|u \in Z(\overline{\ell n})] \to k[x_u|u \in Z(\overline{\ell n}), nu \neq 0], \quad x_u \mapsto \begin{cases} b_u\,, \text{ if } & u \in Z(\overline{n}) \\ x_u, \text{ else} \end{cases}.$$

Soit $S = k[y_u, x_v|u \in Z(\overline{n}), v \in Z(\overline{\ell n})]$, nous pouvons considérer J comme un sous-ensemble de S via l'inclusion naturelle de $k[x_u|u \in Z(\overline{\ell n})]$ dans S. Soit $\mathcal{L}'$ l'idéal de S engendré par J avec les éléments $y_u - \sum_{t \in Z(\overline{\ell})} x_{u+t}$.

Proposition 8. En gardant les même notations qu'au dessus, soit $\mathcal{L} = \mathcal{L}' \cap k[y_u|u \in Z(\overline{n})]$. Soit V_0 la sous-variété de dimension 0 de $\mathbb{A}^{n^g}$ définie par l'idéal $\mathcal{L}$. La variété algébrique $\pi_2(\pi_1^{-1}(b_u)_{u \in Z(\overline{n})})$ est isomorphe à une sous-variété de V_0.

4.3.1.3 Un algorithme efficace

La partie la plus difficile du calcul de correspondance modulaire est la résolution d'un certain système algébrique sur un corps k définit par un idéal I. Nous proposons pour cette partie une nouvelle stratégie qui se fonde sur les hypothèses suivantes : nous supposons que $I \subset k\,[x_1, \ldots, x_n]$ est un idéal de dimension zéro. De plus, nous faisons l'hypothèse qu'il est possible de séparer l'ensemble des variables en deux sous-ensemble $[x_1, \ldots, x_n] = [x_1, \ldots, x_k] \cup [x_{k+1}, \ldots, x_n] = X \cup Y$ tels que $J = I \cap k\,[x_{k+1}, \ldots, x_n] = I \cap k\,[Y]$ contient des polynômes de petit degré (cela n'est vrai en général). De manière optionnelle, on peut supposer que

$$\deg\left(\sqrt{I}\right) \ll \deg\left(I\right).$$

Dans la suite, nous donnons la stratégie générale pour résoudre un tel système. Pour les détails, nous renvoyons le lecteur à [35].

Étape 1 En utilisant un algorithme spécifique, nous calculons une base de Groebner tronquée pour un ordre d'élimination et une graduation modifiée. Cela nous permet d'obtenir un idéal de dimension zéro J_1 contenu dans J. En général, J_1 n'est pas égal à J. La sortie de l'algorithme est une suite de polynômes $[p_1, \ldots, p_l]$ dans $k\,[Y]$ tels que J_1 soit engendré par $(p_1, \ldots, p_l)$.

Étape 2 Calculer une base de Groebner G_{DRL} de J_1 pour un ordre du degré total (DRL ou grevlex). Cela peut se faire avec n'importe quel algorithme de calcul efficace de base de Groebner (par exemple F_4).

Étape 3 Calculer une base de Groebner G_{Lex} de J_1 pour un ordre léxicographique. Cela peut se faire avec l'algorithme FGLM en changeant l'ordre monomial de G_{DRL}.

Étape 4 Calculer une décomposition en idéaux premiers de :

$$\sqrt{J_1} = P_1 \cap \cdots \cap P_r$$

Nous supposons que $\deg(P_i) = 1$ (si ce n'est pas le cas, nous remplaçons k par une extension algébrique)

Étape 5 Pour i allant de 1 à r, nous répétons les étapes a,b,c pour l'idéal $(P_i) + I$:

(a) Calculer une base de Groebner G_i de $(P_i) + I$ pour un ordre du degré total (DRL).

(b) Changer l'ordre monomial afin d'obtenir G'_i une base de Groebner pour l'ordre lexicographique de $(P_i) + I$.

(c) Calculer une décomposition en premiers de $\sqrt{(P_i) + I} = P_{j_{i-1}+1} \cap \cdots \cap P_{j_i}$ (par convention $j_{-1} = r$).

À la fin de l'algorithme, nous obtenons une décomposition de l'idéal I

$$\sqrt{I} = P_{r+1} \cap \cdots \cap P_{j_r}$$

4.3.1.4 Conclusion

Nous avons implémenté l'algorithme précédent et nous donnons des exemples de temps de calcul dans [35]. Nous remarquons que cet algorithme en plus de son intérêt général permet d'accélérer la phase d'initialisation de l'algorithme de comptage de points présenté dans [17].

Chapitre 5

Tests d'aléa et théorie de l'information

Dans cette section, nous reprenons les résultats de [82].

Nous fixons les notations qui seront utilisées dans toute la section. Soit Σ un alphabet. Nous notons Σ^* l'ensemble de toutes les suites à coefficients dans Σ. Un élément u de Σ^* est donné par une suite (u_n) d'éléments de Σ indicés par $n \in \mathbb{N}$. De la même manière, pour $k \in \mathbb{N}$, nous notons Σ^k l'ensemble des suites finies de longueur k à coefficients dans Σ. Un élément u de Σ^k s'écrit

$$u = u_0 \dots u_{k-1}$$

avec $u_i \in \Sigma$ pour $i = 0 \dots k-1$. Notons que Σ^k est un ensemble fini qui peut donc être lui-même considéré comme un alphabet.

5.1 Modèle statistique associé à un dispositif

Il est habituel en cryptographie d'utiliser un générateur d'aléa afin de produire une distribution qui se comporte comme une suite de variables aléatoires binaires. Dans cette section, afin de clarifier le lien avec des définitions classiques d'une suite aléatoire souvent citées dans la littérature [56] nous adoptons une approche légèrement différente en disant qu'un générateur aléatoire produit une suite binaire infinie. Cette hypothèse correspond à une idéalisation de la réalité où l'on ferait marcher le générateur pendant une période de temps infinie. À partir d'une suite infinie, il est facile de retrouver une distribution en calculant une probabilité empirique comme expliqué plus loin. Il est aussi facile de produire une suite aléatoire qui se comporte suivant une certaine distribution. Nous voyons donc que les deux points de vue, distribution et suite binaire infinie, sont équivalents.

Nous disons qu'une source d'aléa S_T est une application d'un espace de paramètres T dans l'ensemble Σ^*. L'espace des paramètres T peut-être discret ou discontinu. Dans le cas d'un générateur physique T est un ensemble de variables continues qui décrivent l'état du générateur (la température du circuit, position des balances). Pour un registre à décalage filtré, T est l'espace discret décrivant l'ensemble des vecteurs d'initialisation possibles, le polynôme de rebouclage et la fonction de filtrage.

En pratique, l'ensemble des paramètres prend en compte le fonctionnement normal de la source ainsi que la possibilité de pannes. Il est possible que la source produise des suites ayant de bonnes propriétés statistiques pour certaines valeurs des paramètres dans T et de mauvaises propriétés statistiques pour d'autres valeurs de T. Par exemple, un générateur d'aléa physique pourrait être conçu de manière à ce qu'il produise des bits avec un biais p indépendant des tirages précédents. Il sort “1” avec probabilité p et “0” avec probabilité $q = 1 - p$. Un procédé de fabrication difficile à contrôler peut influencer le paramètre p. En conséquence, il est nécessaire de disposer d'une procédure pour évaluer le générateur afin de rejeter ceux dont le paramètre est loin de $1/2$. Une solution à ce problème est donné par la théorie des tests d'hypothèses.

Nous voudrions aussi avoir une définition d'une suite produite par le fonctionnement normal de la source. Il y a plusieurs définitions d'une suite aléatoire, définitions plus ou moins adaptées au contexte de la cryptographie. La définition la plus générale est due à Kolmogorov [59]. Cette définition repose sur la longueur du plus petit programme capable de produire une certaine suite binaire finie. Dans [89], Martin-Löf montre que toute suite aléatoire au sens de Kolmogorov passe tous les tests statistiques possibles. Il étend aussi la définition de Kolmogorov au cas de suites infinies. Par la suite, nous préférons utiliser une définition plus pratique donnée dans le survol de Knuth [56] et montrons comment adapter cette définition afin de retrouver la propriété d'imprédictibilité habituelle en cryptographie.

Soit W^k l'application de Σ^* dans l'ensemble des suites à coefficients dans Σ^k, qui fait correspondre à $u \in \Sigma^*$ l'unique suite $(w_n) = (W^k(u)_n) \in (\Sigma^k)^*$, telle que

$$u = w_0 \,\|\, w_1 \,\|\, \ldots \,\|\, w_q \,\|\, \ldots$$

avec $\|$ la concaténation.

Une suite d'évènements est définie par une suite $(u_n)_{n\in\mathbb{N}}$ à coefficients dans un ensemble fini Ω. Par exemple, Ω peut être l'alphabet Σ. Pour $x \in \Omega$, la *probabilité empirique* d'un évènement x, notée

$$P_e[(u_n) = x],$$

est définie par la limite suivante si elle existe :

$$\lim_{k\to\infty} \frac{S_k(x)}{k}, \tag{5.1}$$

avec $S_k = \sharp\{n < k | u_n = x\}$.

À une suite binaire infinie, il est possible d'associer pour tout $n \in \mathbb{N}^*$ une distribution de probabilité sur Σ^k donnée par la probabilité empirique de $W^k(u)_n$. En particulier, une source définit une application de l'ensemble des paramètres T dans l'ensemble des distributions de probabilité sur Σ^k pour tout k c'est-à-dire un *modèle statistique* sur Σ^k.

5.2 Définition d'une suite aléatoire

La définition suivante est due à Borel [56].

Définition 10. Pour $l \in \mathbb{N}^*$, une suite $u \in \Sigma^*$ est l-équidistribuée si pour tout $x \in \Sigma^l$, $P_e[(W^l(u)_n) = x] = (\frac{1}{\sharp\Sigma})^l$. Une suite $u \in \Sigma^*$ est ∞-équidistribuée si elle est l-équidistribuée pour tout $l \in \mathbb{N}^*$.

Nous aurons aussi besoins de la définition suivante tirée de [59].

Définition 11. Soit Σ un alphabet. Une "règle de sous-suite" $\mathcal{R}(f)$ ou simplement $\mathcal{R}$ si aucune confusion n'est à craindre est une fonction calculable f de $\cup_{k=1}^{\infty}\Sigma^k$ dans $\{0, 1\}$.

Une telle règle de sous-suite définit une sous-suite $(u_n)\mathcal{R}$ d'une suite infinie (u_n) de la manière suivante : le n-ième terme de (u_n) est dans la suite $(u_n)\mathcal{R}$ si et seulement si $f(u_0, \ldots, u_{n-1}) = 1$.

Par fonction calculable, nous entendons qu'il existe un algorithme f qui prend en entrée un nombre quelconque d'éléments de Σ avec un symbole de terminaison et détermine la valeur de $f(x_0, \ldots, x_{n-1})$. Par exemple, nous pouvons prendre un certain modèle de calcul $\mathcal{M}$ et dire qu'un algorithme est un élément de ce modèle de calcul. Il est de plus possible de supposer que f est pris dans une certaine famille $\mathcal{F}$ d'éléments de $\mathcal{M}$. Cette famille peut être obtenue en ajoutant certaines restrictions à la consommation de temps ou de mémoire de l'algorithme dans le modèle de calcul donné. En particulier, il est possible d'imposer que f soit un algorithme en temps polynomial par rapport à la longueur de l'entrée.

Soit (s_n) une suite de termes de $\{0, 1\}$. La densité $d_{(s_n)}$ de (s_n) est donnée par la limite si elle existe :

$$d_{(s_n)} = \lim_{n\to\infty} \frac{\sum_{i=0}^{n-1} s_i}{n}. \tag{5.2}$$

Si $d_{(s_n)} = 0$ (resp. $d_{(s_n)} > 0$) nous disons que (s_n) est non dense (resp. dense). Maintenant soit $(u_n) \in \Sigma^*$ et $\mathcal{R}_{(f)}$ une règle de sous-suite pour (u_n). Nous disons que $\mathcal{R}$ est non dense (resp. dense) par rapport à (u_n) si la suite $s_n = f(u_0, \ldots, u_{n-1})$ est non dense (resp. dense).

Nous pouvons maintenant donner notre définition d'une suite aléatoire.

Définition 12. Soit $\mathcal{F}$ une famille d'éléments de $\mathcal{M}$ un modèle de calcul. Une suite $(u_n) \in \Sigma^*$ est $R7(\mathcal{F})$ si pour toute *règle de sous-suite dense* $\mathcal{R}_{(f)}$ définie par un certain élément f de $\mathcal{F}$, $(u_n)\mathcal{R}$ est 1-équidistribuée.

5.3 Aléa et imprédictibilité

Afin de justifier cette dernière définition, nous montrons qu'elle contient la notion d'imprédictibilité usuelle en cryptographie (voir [50]). Pour voir cela, considérons $\mathcal{M}$ le modèle de calcul des machines de Turing.

Définition 13. Soit $\mathcal{F}$ une famille d'éléments de $\mathcal{M}$, un $\mathcal{F}$-prédicteur est un algorithme dans $\mathcal{F}$ qui prend en entrée un nombre fini $(u_0, \ldots, u_n)$ d'éléments de Σ et sort un élément de Σ. Afin de prendre en compte le fait que des éléments de $\mathcal{M}$ peuvent avoir une quantité de mémoire limité, nous imposons de plus la restriction suivante pour un $\mathcal{F}$-prédicteur représenté par une machine de Turing : la tête de lecture de la bande contenant les entrées $(u_0, \ldots, u_n)$ ne peut se déplacer que dans une seule direction.

Avec cette dernière hypothèse, nous ne perdons pas de généralité car un algorithme disposant d'assez de mémoire peut copier la bande contenant les entrées sur une autre bande. D'autre part, il est possible d'autoriser un algorithme ayant une quantité de mémoire finie d'avoir accès à une bande infinie.

On définit l'avantage d'un prédicteur P par rapport à la suite $(u_n) \in \Sigma^*$ comme : $\mathrm{Adv}^P((u_n)) = |P_e[P(u_0, \ldots, u_{k-1}) = u_k] - 1/\sharp\Sigma|$.

Définition 14. Soit $(u_n) \in \Sigma^*$, nous disons que (u_n) est $\mathcal{F}$-prédictible (resp. $\mathcal{F}$-imprédictible) s'il existe (resp. s'il n'existe pas) P un $\mathcal{F}$-prédicteur tel que $\mathrm{Adv}^P((u_n)) \neq 0$.

5.4 Un critère lié à l'entropie

Dans cette section, nous donnons une définition pour un test statistique. Nous nous intéressons plus particulièrement au cas des tests statistiques finis et montrons comment associer un modèle statistique à un test statistique fini. Ensuite, nous expliquons comment déduire une certaine classe de prédicteurs à partir d'un test statistique et donnons un critère fondé sur l'entropie pour qu'une suite soit imprédictible.

Nous rappelons que d'après [125], un test statistique n'est rien d'autre qu'un algorithme qui reçoit en entrée une suite binaire finie de longueur n et retourne une suite binaire de longueur $l(n)$ avec $l(n) \leq n$. Un tel algorithme transforme une distribution sur des suites de longueur n en une distribution sur des suites de longueur $l(n)$. À partir de maintenant, afin de simplifier les notations, nous supposons que $\Sigma = \{0, 1\}$ i.e. Σ^* est l'ensemble de toutes les suites binaires.

Partant de la définition d'un automate fini comme dans [52], nous avons la proposition suivante

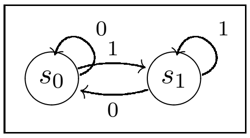
Proposition 9. Il existe une correspondance bijective entre l'ensemble des tests statistiques finis et l'ensemble de automates finis. En conséquence, nous pouvons représenter un test statistique fini F par un triplet (S, f, s_0) avec

- S un ensemble fini d'états $\{s_0, \ldots, s_k\}$,
- $f : S \times \Sigma \mapsto S$ une fonction de transition,
- s_0 un état initial.

La fonction de transition f est définie par : pour $s_i, s_j \in S$ et $\sigma \in \Sigma$, $f(s_i, \sigma) = s_j$ si le test statistique fini passe de l'état s_i à l'état s_j après la lecture de σ sur le ruban d'entrée. À partir de cette donnée, il est possible de calculer l'application $F : \Sigma^{k+1} \to S$ pour tout k avec la formule inductive :

$$F(x_0, \ldots, x_k) = f(F(x_0, \ldots, x_{k-1}), x_k).$$

FIG. 5.1 – Test de fréquence



Un test statistique fini $F = (S, f, s_0)$ peut être représenté par un graphe orienté tel que

- les noeuds du graphe sont les différents états $s_i \in S$,
- les noeuds s_i et s_j sont reliés par une arête avec le label $\sigma \in \Sigma$ si $f(s_i, \sigma) = s_j$.

Nous avons une définition analogue pour les prédicteurs finis.

Définition 15. Un test statistique fini $F = (S, f, s_0)$ (resp. $P = (S, f, s_0, p)$ un prédicteur fini) est dit unifilaire si pour tout σ, σ' éléments distincts de Σ et pour tout $s_i \in S$, nous avons $f(s_i, \sigma) \neq f(s_i, \sigma')$.

Nous remarquons qu'un test statistique fini est unifilaire si et seulement si chaque élément de la chaîne de Markov associée est unifilaire [3, pp. 187]. Il est clair que nous ne perdons pas en généralité si nous supposons à partir de maintenant que tous les tests statistiques sont finis.

Il existe une application que nous notons Φ de l'ensemble des tests statistiques finis dans l'ensemble des chaînes de Markov finies paramétrées par $\Lambda = (\lambda_{s_0}, \ldots, \lambda_{s_k}) \in [0,1]^{k+1}$ que nous notons $\mathcal{M}_F(\Lambda)$. L'application Φ envoie $F = (S, f, s_0)$ avec $S = \{s_0, \ldots, s_k\}$ sur $\mathcal{M}_F(\Lambda) = \{(S, M(\Lambda), Z_0), \Lambda = (\lambda_{s_0}, \ldots, \lambda_{s_k}) \in [0,1]^{k+1}\}$ donné par

- S un ensemble d'états.
- Une matrice de transition $M(\Lambda) = [m_{ij}(\Lambda)]$, $0 \leq i, j \leq k$ avec $m_{ij} = \lambda_{s_i}$ si $f(s_i, 1) = s_j$, $m_{ij} = 1 - \lambda_{s_i}$ si $f(s_i, 0) = s_j$ et $m_{ij} = 0$ sinon. On voit immédiatement que $M(\Lambda)$ est une matrice stochastique i.e. nous avons $\sum_{j=0}^{k} m_{ij} = 1$.
- Une distribution initiale Z_0 qui est la distribution avec la masse totale sur l'état s_0.

Définition 16. Soit $F = (S, f, s_0)$ un test statistique fini. Soit $s_i \in S$, s_i est dit apériodique si le plus grand diviseur commun de l'ensemble des chemins de la représentation graphe de F partant de s_i et revenant sur s_i est égal à 1. Nous disons que F est apériodique si tous ses états sont apériodiques.

Le test statistique F est dit indécomposable s'il n'existe pas de sous-ensemble propre $W \subset S$ tel que $\cup_{\sigma \in \Sigma} f(W, \sigma) \subset W$.

Nous disons que F est ergodique s'il est apériodique et indécomposable.

Définition 17. Soit Λ un ensemble de paramètres et Ω un espace de probabilité. Un *modèle statistique* sur Ω est la donnée d'une distribution de probabilité sur Ω dépendant des paramètres Λ.

Soit $F = (S, f, s_0)$ un test statistique fini ayant pour ensemble d'états $\{s_0, \ldots, s_k\}$ que nous considérons comme un espace de probabilité équidistribué. Soit $\Phi(F) = \{(S, M(\Lambda), s_0), \Lambda \in]0, 1[^{k+1}\}$ la famille des chaînes de Markov associées à F. Ces chaînes de Markov ont un unique état stationnaire par [3, Th. 6.3.3] $W = [w_0, \ldots, w_k]$ qui peut se calculer grâce à la relation de Chapman-Kolmogorov :

$$WM(\Lambda) = W.$$

De cette manière, nous avons associé à S un modèle statistique ayant pour espace de paramètres Λ.

Exemple 1. Le modèle statistique associé au test de fréquence monobit a deux paramètres λ_{s_0} et λ_{s_1} liés par la relation $\lambda_{s_0} + \lambda_{s_1} = 1$ (voir la figure 1). Ce n'est rien d'autre que le modèle d'une source binaire sans mémoire.

Nous pouvons maintenant introduire un invariant qui mesure l'incertitude d'une suite par rapport à un test statistique. Soit $F = (S, f, s_0)$ un test statistique fini avec $S = \{s_0, \ldots, s_k\}$ et $T = (S, M(\Lambda), Z_0) \in \Phi(F)$. Par définition, T donne une suite de variables aléatoires $Z_0, Z_1, \ldots, Z_n, \ldots$ sur l'ensemble des états S. Nous définissons l'entropie de T comme la limite :

$$H(T) = \lim_{n\to\infty} \frac{H(Z_0, \ldots, Z_n)}{n+1}, \tag{5.3}$$

où H, est la fonction entropie usuelle donnée par

$$H(Z_0, \ldots, Z_n) = -\sum_{0 \le i_0, \ldots, i_n \le k} p(z_{0,i_0}, \ldots, z_{n,i_n}) \log(p(z_{0,i_0}, \ldots, z_{n,i_n}))$$

avec $p(z_{0,i_0}, \ldots, z_{n,i_n}) = P[Z_0 = s_{i_0}, \ldots, Z_n = s_{i_n}]$.

Nous pouvons alors énoncer la proposition

Proposition 10. Soit $\mathcal{M}_0$ l'ensemble des machines de Turing finies, alors une suite (u_n) est dans $R7(\mathcal{M}_0)$ si et seulement si pour tout test statistique fini F, $H((u_n)(F)) = 1$.

5.5 Une classification des tests statistiques finis

Dans cette section, nous définissons une relation d'ordre sur l'ensemble des tests statistiques finis. De manière imprécise, si F et F' sont des tests statistiques finis et si F est plus fort que F' alors toute suite qui passe le test F passe aussi le test F'.

Définition 18. Soient $F = (S, f, s_0)$ et $F' = (S', f', s'_0)$ deux tests statistiques finis et ergodiques. Un morphisme de F dans F' est une application $\chi : S \to S'$ telle que $\chi(s_0) = s'_0$ et compatible avec les fonctions de transition i.e. pour tout $s \in S$ et $\sigma \in \Sigma$, nous avons

$$\chi(f(s, \sigma)) = f'(\chi(s), \sigma).$$

Un morphisme $\chi : F \to F'$ de tests statistiques finis ergodiques induit une application $\chi_M : \Phi(F) \to \Phi(F')$ sur la famille associée de chaînes de Markov définie comme suit : si $\Phi(F) = (S, M(\Lambda), s_0)$ et $\Phi(F') = (S', M'(\Lambda'), t_0)$, alors nous posons

$$\chi_M((S, M(\lambda_{s_0}, \ldots, \lambda_{s_k}), s_0)) = (S', M'(\lambda'_{t_0}, \ldots, \lambda'_{t_l}), t_0)$$

avec pour $i = 0, \ldots, l$,

$$\Big(\sum_{s_j \in \chi^{-1}(t_i)} w_{s_j}\Big)\lambda'_{t_i} = \sum_{s_j \in \chi^{-1}(t_i)} w_{s_j} \lambda_{s_j}, \tag{5.4}$$

où $[w_{s_0}, \ldots, w_{s_k}]$ est l'état stationnaire de probabilité de $(S, M((\lambda_{s_0}, \ldots, \lambda_{s_k})), s_0)$.

Proposition 11. Soit $\chi : F \to F'$, un morphisme de tests statistiques et χ_M le morphisme associé sur les chaînes de Markov. Soit $(S, M(\Lambda), s_0)$ un élément de la famille $\Phi(F)$ alors

$$H((S, M(\Lambda), s_0)) \leq H(\chi_M(S, M(\Lambda), s_0)).$$

Pour une preuve voir [82].

Définition 19. Soient F et F' deux tests statistiques finis. S'il existe un morphisme $\chi : F \to F'$, nous disons que F est plus fort que F' et nous notons $F \geq F'$. Si $F \geq F'$ et $F' \geq F$, nous disons que F et F' sont isomorphes.

Nous avons la facile

Proposition 12. La relation $\geq$ est une relation d'ordre sur l'ensemble $\mathcal{F}$ de tous les tests statistiques finis modulo isomorphisme.

Le corollaire suivant montre que si un test F est plus fort qu'un autre test F' alors toute suite qui passe le test F passe aussi le test F'.

Corollaire 6. Si $\chi : F \to F'$ est une application entre deux tests statistiques finis alors pour toute suite $(u_n) \in \Sigma^*$, $H((u_n)(F)) \leq H((u_n)(F'))$.

5.6 Famille complète de tests

Nous avons défini une certaine relation d'ordre sur l'ensemble des tests statistiques finis. Nous avons vu que si un test F est plus fort que F' alors toute suite qui passe le test F passe aussi le test F'. Une question naturelle est la suivante : existe t-il un test statistique fini qui est plus fort que tous les autres ? Il est facile de répondre par la négative à cette question car un tel test aurait un nombre d'états plus grand que n'importe quel autre test statistique fini ce qui est impossible. On voudrait alors décrire une certaine famille de tests statistiques F_i que nous appelons famille complète de tests telle que si une suite passe tous les tests de F_i alors elle passe tous les tests statistiques finis. Nous allons voir par la suite, qu'une réponse à cette question peut se déduire du travail de Shannon sur la théorie de l'information [113].

Définition 20. Soit $F = (S_i, f_i, s_{0i})$ $i \in I$, une famille de tests statistiques finis. Considérons la fonction de transition $\times_{i \in I} f_i$ sur le produit cartésien de l'ensemble des états $\times_{i \in I} S_i$ définie par $(\times_{i \in I} f_i)((s_i)_{i \in I}, \sigma) = (f_i(s_i, \sigma))_{i \in I}$ avec $(s_i)_{i \in I} \in \times_{i \in I} S_i$ et $\sigma \in \Sigma$. Soit W dans $\times_{i \in I} S_i$ le plus petit ensemble parmi ceux contenant $(s_{0i})_{i \in I}$ et stable par l'action de $\times_{i \in I} f_i$ i.e. nous avons $(\times_{i \in I} f_i)(W) \subset W$.

Si I est fini alors $(W, \times_i f_i, (s_{0i})_{i \in I})$ est un test statistique fini appelé le produit de $(F_i)_{i \in I}$ et noté $\times_{i \in I} F_i$.

Soit $F = (S_i, f_i, s_{0i})$, $i \in I$ une famille finie de tests statistiques finis. Si nous ne supposons pas que I est fini alors $\times_{i \in I} F_i$ n'est plus en général un test statistique fini puisque nous obtenons un automate ayant un nombre infini d'états. Néanmoins, il est possible de considérer $\times_{i \in I} F_i$ comme un graphe orienté infini. Nous avons alors le

Théorème 17. Soit $(F_i), i \in I$ une famille de tests statistiques finis. Si $\times_{i \in I} F_i$ est un arbre, i.e. un graphe sans cycle alors pour toute suite binaire (u_n), et tout test statistique fini F, nous avons

$$H((u_n)(F)) \geq \inf\{H((u_n)(F_i)), i \in I\}.$$

Pour une preuve de ce théorème, voir [82].

5.7 Conclusion

D'une manière générale, la pratique des tests statistiques afin de vérifier le caractère aléatoire d'un dispositif se caractérise par

- le choix d'un modèle statistique adapté au dispositif ;
- le choix d'un ensemble restreint de tests statistiques associés au modèle statistique.

En pratique, le choix du modèle statistique utilisé pour décrire le dispositif se base sur des hypothèses heuristiques. L'expérience montre que la plupart du temps, les problèmes survenant dans la conception d'un dispositif affecteront le comportement d'une manière bien précise : par exemple, par la duplication de certains bits, la répétition de certains motifs à intervalles périodiques. Ce genre de comportement non aléatoire peut être facilement identifié en utilisant le test de fréquence, le test de Maurer ou le test de collision. Cependant, si un de ces tests détecte une déviation, il est difficile de décider si cela est du à une panne réelle en l'absence d'explication systémique.

Bibliographie

[1] Leonard M. Adleman, Jonathan DeMarrais, and Ming-Deh Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In *Algorithmic number theory (Ithaca, NY, 1994)*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 28–40. Springer, Berlin, 1994.

[2] A. V. Aho, K. Steiglitz, and J. D. Ullman. Evaluating polynomials at fixed sets of points. *SIAM J. Comput.*, 4(4) :533–539, 1975.

[3] Robert B. Ash. *Information theory.* Dover Publications Inc., New York, 1990. Corrected reprint of the 1965 original.

[4] D. Bernstein. Curve25519 : new Diffie-Hellman speed records. In M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, editors, *PKC 2006*, volume 3958 of *Lecture Notes in Comput. Sci.*, pages 207–228. Springer-Verlag, 2006.

[5] D. Bernstein and T. Lange. Faster addition and doubling on elliptic curves. In K. Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Comput. Sci.*, pages 29–50. Springer-Verlag, 2007.

[6] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Proc. of IEEE Symposium on Security and Privacy*, pages ?–?, 2007.

[7] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *Advances in cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Comput. Sci.*, pages 223–238. Springer, Berlin, 2004.

[8] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Advances in cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Comput. Sci.*, pages 56–73. Springer, Berlin, 2004.

[9] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Proc. of Advances in Cryptology – Eurocrypt'05*, pages 440–456, 2005.

[10] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Proc. of Advances in Cryptology – Crypto'05*, pages 258–275, 2005.

[11] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma Algebra System I : The User Language. *J. Symbolic Comp.*, 24(3) :235–265, 1997.

[12] A. Bostan, F. Chyzak, F. Ollivier, B. Salvy, É. Schost, and A. Sedoglavic. Fast computation of power series solutions of systems of differential equations. In *SODA*, pages 1012–1021, 2007.

[13] Alin Bostan, Bruno Salvy, and Éric Schost. Power series composition and change of basis. In David J. Jeffrey, editor, *ISSAC'08*. ACM Press, To appear. Proceedings of ISSAC'08, Hagenberg, Austria.

[14] D. Brown. Generic groups, collision resistance, and ecdsa. In *Designe, Codes and Cryptography*, volume 35, pages 119–152. 2005.

[15] D. Brown. On the provable security of ecdsa. In I. Blake and G. Seroussi, editors, *Advances in Elliptic Curve Cryptography*, pages 21–40. Cambridge University Press, 2005.

[16] David G. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Math. Comp.*, 48(177) :95–101, 1987.

[17] D. Carls, R. and Lubicz. A p-adic quasi-quadratic time and quadratic space point counting algorithm. 2008. preprint.

[18] R. Carls. Canonical coordinates on the canonical lift. *J. Ramanujan Math. Soc.*, 22(1) :1–14, 2007.

[19] R. Carls, D. Kohel, and D. Lubicz. Higher dimensional 3-adic CM construction. *J. Algebra*, 319 :971–2006, 2008.

[20] R. Carls and D. Lubicz. Magma implementation of the genus 1 point counting algorithm, 2007. Available at `http ://www.mathematik.uni-ulm.de/ReineMath/mitarbeiter/carls/`.

[21] Robert Carls. Canonical coordinates on the canonical lift. *J. Ramanujan Math. Soc.*, 22(1) :1–14, 2007.

[22] D. Chatel, G. and Lubicz. A point counting algorithm using cohomology with compact support. 2008. preprint.

[23] D. V. Chudnovsky and G. V. Chudnovsky. Sequences of numbers generated by addition in formal groups and new primality and factorization tests. *AAM*, 7 :385–434, 1986.

[24] H. Cohen and G. Frey, editors. *Handbook of elliptic and hyperelliptic curve cryptography.* Chapman & Hall / CRC, 2005.

[25] Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptology*, 10(4) :233–260, 1997.

[26] Jean-Sébastien Coron and David Naccache. An accurate evaluation of Maurer's universal test. In *Selected areas in cryptography (Kingston, ON, 1998)*, volume 1556 of *Lecture Notes in Comput. Sci.*, pages 57–71. Springer, Berlin, 1999.

[27] Jean-Marc Couveignes and Thierry Henocq. Action of modular correspondences around CM points. In *Algorithmic number theory (Sydney, 2002)*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 234–243. Springer, Berlin, 2002.

[28] M. Demazure. *Lectures on p-divisible groups.* Number 302 in LNM. Springer, 1972.

[29] Jan Denef and Frederik Vercauteren. An extension of Kedlaya's algorithm to Artin-Schreier curves in characteristic 2. In *Algorithmic number theory (Sydney, 2002)*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 308–323. Springer, Berlin, 2002.

[30] Jan Denef and Frederik Vercauteren. Counting points on C_{ab} curves using Monsky-Washnitzer cohomology. *Finite Fields Appl.*, 12(1) :78–102, 2006.

[31] Jan Denef and Frederik Vercauteren. An extension of Kedlaya's algorithm to hyperelliptic curves in characteristic 2. *J. Cryptology*, 19(1) :1–25, 2006.

[32] Alexander W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In *Advances in cryptology—ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Comput. Sci.*, pages 100–109. Springer, Berlin, 2002.

[33] Noam D. Elkies. Elliptic and modular curves over finite fields and related computational issues. In *Computational perspectives on number theory (Chicago, IL, 1995)*, volume 7 of *AMS/IP Stud. Adv. Math.*, pages 21–76. Amer. Math. Soc., Providence, RI, 1998.

[34] Renée Elkik. Solutions d'équations à coefficients dans un anneau hensélien. *Ann. Sci. École Norm. Sup. (4)*, 6 :553–603 (1974), 1973.

[35] J.-C. Faugère and D. Lubicz. Computing modular correspondance for abelian varieties. 2008. preprint.

[36] John D. Fay. *Theta functions on Riemann surfaces.* Springer-Verlag, Berlin, 1973. Lecture Notes in Mathematics, Vol. 352.

[37] Amos Fiat and Moni Naor. Broadcast encryption. In *Proc. of Advances in Cryptology – Crypto'93*, pages 480–491, 1993.

[38] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Proc. of Advances in Cryptology – PKC'99*, pages 53–68, 1999.

[39] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Proc. of Advances in Cryptology – Crypto'99*, pages 537–554, 1999.

[40] P. Gaudry. A comparison and a combination of SST and AGM algorithms for counting points of elliptic curves in characteristic 2. In *Advances in cryptology—ASIACRYPT 2002*, Lecture Notes in Comput. Sci. Springer, Berlin, December 2002.

[41] P. Gaudry. Fast genus 2 arithmetic based on Theta functions. *J. of Mathematical Cryptology*, 1 :243–265, 2007.

[42] P. Gaudry, T. Houtmann, D. Kohel, C. Ritzenthaler, and A. Weng. The p-adic cm method for genus 2 curves. Available at `http ://arxiv.org/abs/math.NT/0503148`, 2005.

[43] P. Gaudry and D. Lubicz. The arithmetic of characteristic 2 kummer surfaces. 2008. preprint.

[44] P. Gaudry and É. Schost. Construction of secure random curves of genus 2 over prime fields. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Comput. Sci.*, pages 239–256. Springer-Verlag, 2004.

[45] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. A double large prime variation for small genus hyperelliptic index calculus. *Math. Comp.*, 76(257) :475–492 (electronic), 2007.

[46] P. Gaudry and E. Thomé. The mpFq library and implementing curve-based key exchanges. In *SPEED : Software Performance Enhancement for Encryption and Decryption*, pages 49–64, 2007.

[47] Pierrick Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. In *Advances in cryptology—EUROCRYPT 2000 (Bruges)*, volume 1807 of *Lecture Notes in Comput. Sci.*, pages 19–34. Springer, Berlin, 2000.

[48] Pierrick Gaudry. Cardinality of a genus 2 hyperelliptic curve over $\mathrm{GF}(5\cdot 10^{24}+41)$. Email at the Number Theory List, September 2002.

[49] Pierrick Gaudry and Nicolas Gürel. An extension of Kedlaya's point-counting algorithm to superelliptic curves. In *Advances in cryptology—ASIACRYPT 2001 (Gold Coast)*, volume 2248 of *Lecture Notes in Comput. Sci.*, pages 480–494. Springer, Berlin, 2001.

[50] Oded Goldreich. *Foundations of cryptography.* Cambridge University Press, Cambridge, 2001. Basic tools.

[51] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. of ACM-CCS'06*, pages 89–98, 2006.

[52] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, And Computation.* Addison Wesley Longman, 2006.

[53] Antoine Joux. The Weil and Tate pairings as building blocks for public key cryptosystems. In *Algorithmic number theory (Sydney, 2002)*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 20–32. Springer, Berlin, 2002.

[54] K.S. Kedlaya. Counting points on hyperelliptic curves using Monsky Washnitzer cohomology. *Journal of the Ramanujan Mathematical Society*, 16 :323–328, 2001.

[55] Hae Young Kim, Jung Youl Park, Jung Hee Cheon, Je Hong Park, Jae Heon Kim, and Sang Geun Hahn. Fast Elliptic Curve Point Counting Using Gaussian Normal Basis. In Claus Fieker and David R. Kohel, editors, *Algorithmic Number Theory, 5th International Symposium, ANTS-V*, pages 292–307, Berlin, July 2002. Springer Verlag.

[56] Donald E. Knuth. *The art of computer programming. Vol. 2.* Addison-Wesley Publishing Co., Reading, Mass., second edition, 1981. Seminumerical algorithms, Addison-Wesley Series in Computer Science and Information Processing.

[57] Neal Koblitz and Alfred Menezes. Another look at generic groups. *Adv. Math. Commun.*, 1(1) :13–28, 2007.

[58] David R. Kohel. The AGM-$X_0(N)$ Heegner point lifting algorithm and elliptic curve point counting. In *Advances in cryptology—ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Comput. Sci.*, pages 124–136. Springer, Berlin, 2003.

[59] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Internat. J. Comput. Math.*, 2 :157–168, 1968.

[60] Y. Laszlo and C. Pauly. The action of the Frobenius maps on rank 2 vector bundles in characteristic 2. *J. Algebraic Geom.*, 11(2) :219–243, 2002.

[61] Y. Laszlo and C. Pauly. The Frobenius map, rank 2 vector bundles and Kummer's quartic surface in characteristic 2 and 3. *Adv. Math.*, 185(2) :246–269, 2004.

[62] Alan G. B. Lauder. Computing zeta functions of Kummer curves via multiplicative characters. *Found. Comput. Math.*, 3(3) :273–295, 2003.

[63] Alan G. B. Lauder. Counting solutions to equations in many variables over finite fields. *Found. Comput. Math.*, 4(3) :221–267, 2004.

[64] Alan G. B. Lauder. Deformation theory and the computation of zeta functions. *Proc. London Math. Soc. (3)*, 88(3) :565–602, 2004.

[65] Alan G. B. Lauder. Rationality and meromorphy of zeta functions. *Finite Fields Appl.*, 11(3) :491–510, 2005.

[66] Alan G. B. Lauder. Rigid cohomology and p-adic point counting. *J. Théor. Nombres Bordeaux*, 17(1) :169–180, 2005.

[67] Alan G. B. Lauder. A recursive method for computing zeta functions of varieties. *LMS J. Comput. Math.*, 9 :222–269 (electronic), 2006.

[68] Alan G. B. Lauder and Daqing Wan. Computing zeta functions of Artin-Schreier curves over finite fields. *LMS J. Comput. Math.*, 5 :34–55 (electronic), 2002.

[69] Alan G. B. Lauder and Daqing Wan. Computing zeta functions of Artin-Schreier curves over finite fields. II. *J. Complexity*, 20(2-3) :331–349, 2004.

[70] Bernard Le Stum. Filtration par le poids sur la cohomologie de de Rham d'une courbe projective non singulière sur un corps ultramétrique complet. *Rend. Sem. Mat. Univ. Padova*, 93 :43–85, 1995.

[71] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261 :513–534, 1982.

[72] Reynald Lercier and David Lubicz. Cardinality of a genus 2 hyperelliptic curve over $GF(2^{4001})$. Email at the Number Theory List, 2002.

[73] Reynald Lercier and David Lubicz. Elliptic curve point counting, 100002 bits. Email at the Number Theory List, 2002.

[74] Reynald Lercier and David Lubicz. Elliptic curve point counting, 100002 bits, 2002.

[75] Reynald Lercier and David Lubicz. Elliptic curve point counting, 65538 bits. Email at the Number Theory List, 2002.

[76] Reynald Lercier and David Lubicz. Genus 2 hyperelliptic curve point counting, 16420 bits. Email at the Number Theory List, 2002.

[77] Reynald Lercier and David Lubicz. Counting Points on Elliptic Curves over Finite Fields of Small Characteristic in Quasi Quadratic Time. In Eli Biham, editor, *Advances in Cryptology—EUROCRYPT '2003*, Lecture Notes in Computer Science. Springer-Verlag, May 2003.

[78] Reynald Lercier and David Lubicz. Genus 2 hyperelliptic curve pointcounting, 32770 bits. Email at the Number Theory List, 2003.

[79] Reynald Lercier and David Lubicz. A quasi quadratic time algorithm for hyperelliptic curve point counting. *Ramanujan J.*, 12(3) :399–423, 2006.

[80] David Lubicz. Une description de la cohomologie du complément à un diviseur non réductible de $\mathbf{P}^2$. *Bull. Sci. Math.*, 124(6) :447–458, 2000.

[81] David Lubicz. Sur les pinceaux de courbes définis par une fonction régulière. *Bull. Sci. Math.*, 125(2) :139–160, 2001.

[82] David Lubicz. On a classification of finite statistical tests. *Adv. Math. Commun.*, 1(4) :509–524, 2007.

[83] David Lubicz and Thomas Sirvent. Attribute-based broadcast encryption scheme made efficient. In Serge Vaudenay, editor, *Advances in Cryptology—AFRICACRYPT '2008*, Lecture Notes in Computer Science, pages 325–342. Springer-Verlag, 2008.

[84] David Lubicz and Thomas Sirvent. Pseudo-random groups and related problems. 2008.

[85] J. Lubin, J.-P. Serre, and J. Tate. Elliptic curves and formal groups. Available at `http ://ma.utexas.edu/users/voloch/lst.html`, 1964.

[86] Michael Luby. *Pseudorandomness and cryptographic applications*. Princeton Computer Science Notes. Princeton University Press, Princeton, NJ, 1996.

[87] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2) :373–386, 1988.

[88] George Marsaglia. Diehard. Available at `http ://www.csis.hku.hk/ diehard/`.

[89] Per Martin-Löf. The definition of random sequences. *Information and Control*, 9 :602–619, 1966.

[90] Ueli M. Maurer. A universal statistical test for random bit generators. *J. Cryptology*, 5(2) :89–105, 1992.

[91] William Mendenhall and Richard L. Scheaffer. *Mathematical statistics with applications*. Duxbury Press, North Scituate, Mass., 1973.

[92] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. CRC Press Series on Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, 1997. With a foreword by Ronald L. Rivest.

[93] Jean-François Mestre. Lettre à Gaudry et Harley, 2001. Available at `http ://www.math.jussieu.fr/mestre`.

[94] Jean-François Mestre. Notes of a talk given at the seminar of cryptography of Rennes, 2002. Available at `http ://www.math.univ-rennes1.fr/crypto/2001-02/mestre.ps`.

[95] D. Mumford. On the equations defining abelian varieties. I. *Invent. Math.*, 1 :287–354, 1966.

[96] D. Mumford. On the equations defining abelian varieties. II. *Invent. Math.*, 3 :75–135, 1967.

[97] D. Mumford. *Abelian varieties.* Tata Institute of Fundamental Research Studies in Mathematics, No. 5. Published for the Tata Institute of Fundamental Research, Bombay, 1970.

[98] David Mumford. *Tata lectures on theta II*, volume 43 of *Progress in Mathematics*. Birkhäuser Boston Inc., Boston, MA, 1984. Jacobian theta functions and differential equations, With the collaboration of C. Musili, M. Nori, E. Previato, M. Stillman and H. Umemura.

[99] Dalit Naor, Moni Naor, and Lotspiech. Revocation and tracing schemes for stateless receivers. In *Proc. of Advances in Cryptology – Crypto'01*, pages 41–62, 2001.

[100] V. I. Nechaev. On the complexity of a deterministic algorithm for a discrete logarithm. *Mat. Zametki*, 55(2) :91–101, 189, 1994.

[101] Phong Q. Nguyen and Damien Stehlé. Floating-point LLL revisited. In *Advances in cryptology—EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Comput. Sci.*, pages 215–233. Springer, Berlin, 2005.

[102] NIST. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Available at `http ://csrc.nist.gov/CryptoToolkit/tkrng.html`.

[103] Tatsuaki Okamoto and David Pointcheval. React : Rapid enhanced-security asymmetric cryptosystem transform. In *Proc. of Cryptographers' Track, RSA Conference – CT-RSA'01*, pages 159–175, 2001.

[104] Christos H. Papadimitriou. *Computational complexity.* Addison-Wesley Publishing Company, Reading, MA, 1994.

[105] Michael S. Paterson and Larry J. Stockmeyer. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. Comput.*, 2 :60–66, 1973.

[106] J. Pila. Frobenius maps of abelian varieties and finding roots of unity in finite fields. *Math. Comp.*, 55(192) :745–763, 1990.

[107] Stephen C. Pohlig and Martin E. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. *IEEE Trans. Information Theory*, IT-24(1) :106–110, 1978.

[108] J. M. Pollard. Monte Carlo methods for index computation (mod p). *Math. Comp.*, 32(143) :918–924, 1978.

[109] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Proc. of Advances in Cryptology – Eurocrypt'05*, pages 457–473, 2005.

[110] T. Satoh, B. Skjernaa, and Y. Taguchi. Fast computation of canonical lifts of elliptic curves and its application to point counting. *Finite Fields and Their Applications*, 9(1) :89–101, 2003.

[111] Takakazu Satoh. The canonical lift of an ordinary elliptic curve over a finite field and its point counting. *J. Ramanujan Math. Soc.*, 15(4) :247–270, 2000.

[112] R. Schoof. Counting points on elliptic curves over finite fields. *J. Théorie des nombres de Bordeaux*, 7 :483–494, 1998.

[113] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27 :379–423, 623–656, 1948.

[114] G. Shimura. *Abelian varieties with complex multiplication and modular functions.* Princ. Univ. Press, NJ, 1998.

[115] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in cryptology—EUROCRYPT '97 (Konstanz)*, volume 1233 of *Lecture Notes in Comput. Sci.*, pages 256–266. Springer, Berlin, 1997.

[116] Joseph H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1986. Corrected reprint of the 1986 original.

[117] Joseph H. Silverman. *Advanced topics in the arithmetic of elliptic curves*, volume 151 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1994.

[118] M. Stam. On Montgomery-like representations for elliptic curves over $GF(2^k)$. In Y. Desmedt, editor, *PKC 2003*, volume 2567 of *Lecture Notes in Comput. Sci.*, pages 240–254. Springer-Verlag, 2003.

[119] J. Stern, D. Pointcheval, J. Malone-Lee, and N Smart. Flaws in applying proof methodologies to signature schemes. In *Advances in cryptology—CRYPTO 2002*, volume 2442 of *Lecture Notes in Comput. Sci.*, pages 93–110. Springer, Berlin, 2002.

[120] John Tate. Endomorphisms of abelian varieties over finite fields. *Invent. Math.*, 2 :134–144, 1966.

[121] Nicolas Thériault. Index calculus attack for hyperelliptic curves of small genus. In *Advances in cryptology—ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Comput. Sci.*, pages 75–92. Springer, Berlin, 2003.

[122] Nobuo Tsuzuki. On the Gysin isomorphism of rigid cohomology. *Hiroshima Math. J.*, 29(3) :479–527, 1999.

[123] Paul van Wamelen. Examples of genus two CM curves defined over the rationals. *Math. Comp.*, 68(225) :307–320, 1999.

[124] Frederik Vercauteren, Bart Preneel, and Joos Vandewalle. A memory efficient version of Satoh's algorithm. In *Advances in cryptology—EUROCRYPT 2001 (Innsbruck)*, volume 2045 of *Lecture Notes in Comput. Sci.*, pages 1–13. Springer, Berlin, 2001.

[125] Andrew C. Yao. Theory and applications of trapdoor functions. *IEEE*, pages 80–91, 1982.

Résumé

La cryptographie à clef publique, qui fut inventée dans les années soixante-dix par W. Diffie et M. Hellman, apporte par rapport à la cryptographie symétrique un certain nombre de fonctionnalités particulièrement intéressantes pour les applications pratiques. Sa mise en œuvre repose le plus souvent sur la difficulté calculatoire de certains problèmes issus de la théorie des nombres. De là, on peut déduire des fonctions à sens unique, des fonctions trappes puis construire et prouver par réduction des protocoles permettant de répondre à des objectifs de sécurité variés, les plus courants étant le chiffrement ou la signature numérique.

Un problème classiquement utilisé en cryptographie asymétrique est le problème du logarithme discret qui est par exemple à la base de toute la cryptographie sur courbe elliptique. Le problème du logarithme discret permet de construire des fonctions supposées à sens unique à partir de familles de groupes disposant d'un certain nombre de bonnes propriétés. Dans ce mémoire, nous présentons des techniques permettant de définir, représenter et calculer des familles de groupes utilisables dans des cryptosystèmes à base de logarithme discret. Des considérations de sécurité ou de performance nous amènent à revisiter d'un point de vue algorithmique des concepts développés dans les années soixante pour les besoins de la théorie des nombres et de la géométrie arithmétique : citons par exemple la multiplication complexe, les fonctions thêta algébriques, la cohomologie rigide, la théorie de Serre-Tate.

Abstract

Asymmetric cryptography, invented in the seventies by W. Diffie and M. Hellman, brings a large number of features to the more classical symmetric cryptography which are particularly interesting for practical applications. Most of the time, its operation relies on the computational hardness of some problems coming from number theory. From this, it is possible to deduce one-way functions, trap-door functions and then design and prove by reduction protocols to achieve a variety of security objectives among which the most common are encryption or digital signature.

A problem classically used in asymmetric cryptography is the discrete logarithm problem which is for instance a corner stone for all the elliptic curve cryptography. The discrete logarithm problem allows one to design supposed to be one-way functions from a family of groups enjoying certain good properties. In the dissertation, we present some techniques to define, represent and compute family of groups to be used in discrete logarithm based cryptosystems. Security and performance issues leads us to revisit from a computational point of view some concepts developed during the sixties for the purpose of number theory and arithmetic geometry : complex multiplication, algebraic theta functions, rigid cohomology and Serre-Tate theory are worth mentioning.